IFW
BAC

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Xavier Leroy
Serial No. : 10/069,670
Filed : February 22, 2002
Title : A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM

Art Unit : 2145
Examiner : Jeffrey R. Swearingen

MAIL STOP PETITIONS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION UNDER 37 C.F.R. §1.181(A)

Pursuant to 37 C.F.R. §1.181, Applicant hereby petitions for withdrawal of the holding of abandonment of U.S. Patent Application Serial No. 10/069,670.

Applicant respectfully submits the following facts in support of this petition:

1. On February 22, 2002, the applicant's attorney of record, Michael L. Kenaga of Piper Rudnick (now DLA Piper Rudnick Gray Cary US LLP), filed the above-referenced patent application with the United States Patent and Trademark Office.

2. On May 16, 2005, the Examiner mailed a first Office Action for the above-referenced patent application. The docketing clerk for DLA Piper Rudnick Gray Cary US LLP received such Office Action on May 24, 2005 (See Exhibit A).

3. On September 29, 2005, Mr. Kenaga filed a response to the Office Action of May 16, 2005 for the above-referenced application (See Exhibit B). A Petition for Extension of Time, pursuant to 37 C.F.R § 1.136(a), was filed therewith (See Exhibit C).

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

December 22, 2006
Date of Deposit

Signature

Toni M. Sousa
Typed or Printed Name of Person Signing Certificate

Documentation of receipt by the United States Patent and Trademark Office is enclosed herewith (See Exhibit D).

4. On October 31, 2005, Mr. Kenaga filed a Supplemental Amendment (See Exhibit E) to correct certain typographical errors in the Amendment filed September 29, 2005, which was submitted in the response to the Office Action mailed May 16, 2005. Documentation of receipt by the United States Patent and Trademark Office is enclosed herewith (See Exhibit F).

5. On January 23, 2006, the Examiner mailed a Notice of Non-Compliant Amendment, pursuant to 37 CFR § 1.121, for failure to include a substitute specification in either the initial amendment or supplemental amendment (See Exhibit G). As evidenced by the Declaration of Olga Melendez (See Exhibit H), the Notice of Non-Compliant Amendment was never received by DLA Piper Rudnick Gray Cary US LLP.

6. On February 10, 2006, Grossman, Tucker, Perreault & Pfleger, PLLC received a copy of a duly executed Revocation of Power of Attorney With New Power of Attorney form (See Exhibit I). Such form assigned prosecution of the above-referenced application to the practitioners of Grossman, Tucker, Perreault & Pfleger, PLLC.

7. On February 16, 2006, the above-referenced application was transferred from DLA Piper Rudnick Gray Cary of Chicago, IL to Grossman, Tucker, Perreault & Pfleger, PLLC of Manchester, NH. A copy of the Request to Transfer File letter is enclosed herewith (See Exhibit J).

8. On February 23, 2006, the prosecution regarding the above-referenced application went abandoned due to a failure to respond to the Notice of Non-Compliant Amendment of January 23, 2006.

9. In April of 2006, Mr. Colandreo left Grossman, Tucker, Perreault & Pfleger, PLLC and joined Holland & Knight LLP. In June of 2006, the above-referenced patent

application was transferred to Holland & Knight for continued prosecution by Mr. Colandreo.

10. On August 17, 2006, Mr. Colandreo was notified that the above referenced application had gone abandoned. Upon learning of the abandonment, Mr. Colandreo inspected the file jacket for the subject application and confirmed that such Notice of Non-Compliant Amendment dated January 23, 2006 was not present in the file jacket of the subject application. Mr. Colandreo then submitted a request (See Exhibit K) to the United States Patent and Trademark Office that the Examiner send a copy of the Notice of Non-Compliant Amendment dated January 23, 2006 (See Exhibit G). The Revocation of Power of Attorney with New Power of Attorney form, dated February 9, 2006, was also submitted with Mr. Colandreo's request.

11. On October 23, 2006, Mr. Colandreo received the Declaration of Olga Melendez (Exhibit H). On December 20, 2006, Mr. Colandreo received the docket report from DLA Piper Rudnick necessary for filing this petition (See Exhibit L).

CONCLUSION

It is believed that the above stated facts and the attached documents support this Petition to Revive pursuant to 37 C.F.R. § 1.181. Any delay in filing this petition was due to a lack of notice that the application had gone abandoned and the subsequent time spent in obtaining the documents necessary for filing this petition. Applicant respectfully requests that the Notice of Non-Compliant Amendment dated January 23, 2006 be re-mailed pursuant to the holding in *Delgar v. Schulyer*, 172 USPQ 513 (D.D.C. 1971), as discussed in MPEP § 711.03(c). In the event that the Director determines that a Petition to Revive pursuant to 37 C.F.R. § 1.181 is improper, Applicant is prepared to submit a Petition to Revive Due to Unintentional Delay pursuant to 37 C.F.R. § 1.137(b).

It is believed that no fee is due at this time. In the event that the Patent Office believes that a fee is required to be submitted with this petition, please charge any fee(s) or credit overpayments to Deposit Account 50-2324.

Respectfully submitted,

Date: 22 December 2006

Elizabeth R. Burkhard
Reg. No. 58,710

Holland & Knight LLP
10 St. James Avenue
Boston, MA 02116
Telephone: (617) 573-5850
Facsimile: (617) 523-6850

4072593_v1

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/069,670	02/22/2002	Xavier Leroy	P-6451	3622

28465 7590 05/16/2005

218728-123

DLA PIPER RUDNICK GRAY CARY US LLP
P. O. BOX 64807
CHICAGO, IL 60664-0807

EXAMINER	
SWEARINGEN, JEFFREY R	

ART UNIT	PAPER NUMBER
2145	

DATE MAILED: 05/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Rsp Due 8/16/05

3 month Ext 11/16/05

Office Action Summary	Application No.	Applicant(s)	
	10/069,670	LEROY, XAVIER	
	Examiner	Art Unit	
	Jeffrey R. Swearingen	2145	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☒ Claim(s) *1-27* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 February 2002* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *9/23/02*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "15" has been used to designate both an EEPROM and a serial link between Figure 1a and the specification (page 1). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Figure 1a, item 13; Figure 2, item 103; Figure 3d, item 306; Figure 6, item 16. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

3. A substitute specification in proper idiomatic English and in compliance with 37 CFR 1.52(a) and (b) is required. The substitute specification filed must be accompanied by a statement that it contains no new matter.

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

5. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required. The Examiner has attempted to locate an abstract published with the international application under PCT Article 21, but was unable to do so. The Examiner will gladly remove the abstract objection if Applicant points out to Examiner an abstract that was published with the international application under PCT Article 21 that may have been overlooked.

6. The Examiner is including the following to assist Applicant in the creation of a substitute specification.

Content of Specification

(a) Title of the Invention: See 37 CFR 1.72(a) and MPEP § 606. The title of the invention should be placed at the top of the first page of the specification unless the title is provided in an application data sheet. The title of the invention should be brief but technically accurate and descriptive, preferably from two to seven words may not contain more than 500 characters.

(b) Cross-References to Related Applications: See 37 CFR 1.78 and MPEP § 201.11.

(c) Statement Regarding Federally Sponsored Research and Development: See MPEP § 310.

(d) The Names Of The Parties To A Joint Research Agreement: See 37 CFR 1.71(g).

(e) Incorporation-By-Reference Of Material Submitted On a Compact Disc: The specification is required to include an incorporation-by-reference of electronic documents that are to become part of the permanent United States Patent and Trademark Office records in the file of a patent application. See 37 CFR 1.52(e) and MPEP § 608.05. Computer program

listings (37 CFR 1.96(c)), "Sequence Listings" (37 CFR 1.821(c)), and tables having more than 50 pages of text were permitted as electronic documents on compact discs beginning on September 8, 2000.

Or alternatively, Reference to a "Microfiche Appendix": See MPEP § 608.05(a). "Microfiche Appendices" were accepted by the Office until March 1, 2001.

(f) Background of the Invention: See MPEP § 608.01(c). The specification should set forth the Background of the Invention in two parts:

(1) Field of the Invention: A statement of the field of art to which the invention pertains. This statement may include a paraphrasing of the applicable U.S. patent classification definitions of the subject matter of the claimed invention. This item may also be titled "Technical Field."

(2) Description of the Related Art including information disclosed under 37 CFR 1.97 and 37 CFR 1.98: A description of the related art known to the applicant and including, if applicable, references to specific related art and problems involved in the prior art which are solved by the applicant's invention. This item may also be titled "Background Art."

(g) Brief Summary of the Invention: See MPEP § 608.01(d). A brief summary or general statement of the invention as set forth in 37 CFR 1.73. The summary is separate and distinct from the abstract and is directed toward the invention rather than the disclosure as a whole. The summary may point out the advantages of the invention or how it solves problems previously existent in the prior art (and preferably indicated in the Background of the Invention). In chemical cases it should point out in general terms the utility of the invention. If possible, the nature and gist of the invention or the inventive concept should be set forth. Objects of the invention should be treated briefly and only to the extent that they contribute to an understanding of the invention.

(h) Brief Description of the Several Views of the Drawing(s): See MPEP § 608.01(f). A reference to and brief description of the drawing(s) as set forth in 37 CFR 1.74.

(i) Detailed Description of the Invention: See MPEP § 608.01(g). A description of the preferred embodiment(s) of the invention as required in 37 CFR 1.71. The description should be as short and specific as is necessary to describe the invention adequately and accurately. Where elements or groups of elements, compounds, and processes, which are conventional and generally widely known in the field of the invention described and their exact nature or type is not necessary for an understanding and use of the invention by a person skilled in the art, they should not be described in detail. However, where particularly complicated subject matter is involved or where the elements, compounds, or processes may not be commonly or widely known in the field, the specification should refer to another patent or readily available publication which adequately describes the subject matter.

(j) Claim or Claims: See 37 CFR 1.75 and MPEP § 608.01(m). The claim or claims must commence on separate sheet or electronic page (37 CFR 1.52(b)(3)). Where a claim sets forth a plurality of elements or steps, each element or step of the claim should be separated by a line indentation. There may be plural indentations to further segregate subcombinations or related steps. See 37 CFR 1.75 and MPEP § 608.01(i)-(p).

(k) Abstract of the Disclosure: See MPEP § 608.01(f). A brief narrative of the disclosure as a whole in a single paragraph of 150 words or less commencing on a separate sheet

following the claims. In an international application which has entered the national stage (37 CFR 1.491(b)), the applicant need not submit an abstract commencing on a separate sheet if an abstract was published with the international application under PCT Article 21. The abstract that appears on the cover page of the pamphlet published by the International Bureau (IB) of the World Intellectual Property Organization (WIPO) is the abstract that will be used by the USPTO. See MPEP § 1893.03(e).

(l) Sequence Listing, See 37 CFR 1.821-1.825 and MPEP §§ 2421-2431. The requirement for a sequence listing applies to all sequences disclosed in a given application, whether the sequences are claimed or not. See MPEP § 2421.02.

7. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

8. The use of multiple trademarks has been noted in this application. They should be capitalized wherever they appear and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

9. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

10. 35 U.S.C. 112, first paragraph, requires the specification to be written in "full, clear, concise, and exact terms." The specification is replete with terms which are not clear, concise and exact. The specification should be revised carefully in order to comply with 35 U.S.C. 112, first paragraph. Applicant is pointed throughout the specification to numerous sections where the system consists "at least" of a limitation. The Examiner is unable to determine where in the specification the required facets of the invention end and the optional components begin.

11. Applicant is **STRONGLY URGED** to carefully review and amend the specification, drawings, and claims in order to make a good faith attempt at bringing the application close to conformance with current U.S. practice. Applicant is again reminded that no new matter is allowed within the specification.

Claim Objections

12. Claims 8-14, and 20-25 are objected to under 37 CFR 1.75(c) as being in improper form because a multiple dependent claim cannot depend on another multiple dependent claim. See MPEP § 608.01(n). Accordingly, the claims have not been further treated on the merits. Claims 22 and 25 cannot be dependent from more than one independent claim.

13. Claims 1-27 are objected to because of the following informalities: The claims seem to consist entirely of lengthy preambles, with few to no patentable phrases present. Appropriate correction is required.

14. Claim 3 is objected to because of the method of designation of steps that Applicant has chosen to use (a, b', c', ...) is confusing and unclear to one of ordinary skill in the art. The Examiner suggests that Applicant find a different way of stating the steps of claims 1-3 in order to more clearly explain what Applicant is limiting.

15. Claim 4 is objected to because of the method of designation of steps that Applicant has chosen to use (α, β, γ) is confusing and unclear to one of ordinary skill in the art. The Examiner suggests that Applicant find a different way of stating the steps of claim 4.

16. Claims 5-6 are objected to because of the symbols $\perp$ and $\top$ that Applicant has used to designate *types* within the claims.

17. Claims 5, 6, 8, 17, and all claims dependent upon said claims are objected to because of the use of underlining within the claims.

18. Claims 5, 12-15, 17, 18 and 26 are objected because of using dashes to delineate steps and/or items within the claims.

19. Claim 17 is objected to for the use of bullet points within the claim. The Examiner cannot determine how bullet points help the claim, and actually believes their usage confuses the claim here.

Claim Rejections - 35 USC § 101

20. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

21. Claims 1-3 and 23-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 1-3 are directed to a protocol, which is software that is not embodied in a physical media. Claims 23-25 are directed to a computer program product, but this product is not limited to statutory computer readable media.

Claim Rejections - 35 USC § 112

22. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

23. Claims 1-27 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The claim language is an obvious machine translation. The Examiner has consulted with multiple other Examiners in an attempt to ascertain the field of endeavor of the invention. Claim 1, for example, consists of a lengthy preamble followed by a series of unintelligible sentence fragments. The Examiner is unable to give any patentable weight to the phraseology "*in detecting a command for downloading of this program fragment; and, on a positive response to this stage consisting in detecting a downloading command*", and finds it difficult to believe

that one of ordinary skill in the art could ascertain what this phrase means, much less actually enabling said phraseology. Further using claim 1 as an example, the Examiner is unclear on how the following steps happen with the invention: *detecting a command for downloading a program fragment*, what is considered a *positive response*, how is the *object code* read, how is *subjecting the whole of the object code stored temporarily in memory to a verification process* accomplished, how is *execution inhibited*, how is the *downloaded program fragment recorded in a directory*. Another further example is given for claims 5-6, because of the symbols $\perp$ and $\top$ that Applicant has used to designate *types* within the claims. The Examiner is unaware of any computer compiler that will allow usage of such type designations. Additionally, the Examiner points Applicant to the International Preliminary Examination Report, section VIII: Certain observations on the international application, which brings to light the fact that the *stack is empty at each branching instruction and at each branching target* in order for verification to take place. The phrase *updating of the effect of said current instruction on the type stack and the register type table* does not ensure that this takes place, thus making the claimed invention broader than the written description. Unless the claims clearly state that the stack is empty, the Examiner believes that the verification vaguely alluded to in the unclear specification and claims would be replete with machine errors and thereby not enabled.

24. Claims 1-27 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claims 1-27 refer to a *virtual machine*. It is the Examiner's opinion that the references given by Applicant which refer to a *virtual machine* in the background of the invention are not sufficient for the *virtual machine* definition required for the invention. Applicant is advised to clearly define what a *virtual machine* is for the invention.

25. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

26. Claims 1-27 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

27. The claims are generally narrative and indefinite, failing to conform with current U.S. practice. They appear to be a literal translation into English from a foreign document and are replete with grammatical and idiomatic errors.

28. As an example, claim 1 has following a lengthy preamble (bearing no patentable weight) the step of *in detecting a command for downloading of this program fragment; and, on a positive response to this stage consisting in detecting a downloading command.* The Examiner is unable to ascertain what is being claimed with this phrase.

29. Regarding claims 1, 4, and 23-26, the phrase "such as" renders the claim indefinite because it is unclear whether the limitations following the phrase are part of the claimed invention. See MPEP § 2173.05(d).

30. Claim 1 recites the limitation "this stage" in line 16. There is insufficient antecedent basis for this limitation in the claim. Multiple additional claims have antecedent basis problems, but the Examiner believes that pointing out each individual error at this point would be unnecessary considering the extent to which the claims must be examined to determine what is an antecedent basis problem and what is a grammatical translation problem, given the current state of the claims.

31. Claims 5-6 are rejected because of the symbols $\perp$ and $\top$ that Applicant has used to designate *types* within the claims. The Examiner is unclear on this terminology in regard to usage in computer code for object type definition.

32. Claim 17 is rejected because the use of bullet points has made the claim unintelligible and generally indefinite. The Examiner believes that Applicant is attempting to define an order with the multiple dashes and bullet points used, but cannot ascertain any such order that is currently used by one of ordinary skill in the art.

33. Claims 1-27 are rejected as failing to define the invention in the manner required by 35 U.S.C. 112, second paragraph.

34. The claim(s) are narrative in form and replete with indefinite and functional or operational language. The structure which goes to make up the device must be clearly and positively specified. The structure must be organized and correlated in such a manner as to present a complete operative device. The claim(s) must be in one sentence form only. Note the format of the claims in the U.S. patent(s) cited by Applicant in Applicant's submitted Information Disclosure Statement.

35. Applicant is urged to carefully review all claims in an attempt to expedite prosecution of the application. The numerous 112 errors present in the application may not be limited to what the Examiner has been able to discern due to the current state of the claims. The Examiner has made a good faith effort to identify every general type of error currently present in the claims to Applicant, but notes that additional errors such as enablement, written description, and antecedent basis problems may still be present in the claims and obscured by their current state.

36. In general, the state of the disclosure and claims in the instant application preclude a limitation-by-limitation assessment of the claimed invention compared to the prior art. Therefore prior art is applied under 35 U.S.C. §§ 102 and 103 in an attempt to expedite prosecution in anticipation of future amendments rather than strictly based upon the examiner's assumptions. See *In re Steele*, 305 F.2d 859,134 USPQ 292 (CCPA 1962).

Claim Rejections - 35 USC § 102

37. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

38. Claims 1-7, 15-19, and 26-27 as best understood and interpreted by the Examiner are rejected under 35 U.S.C. 102(b) as being unpatentable by Gosling (U.S. Patent No. 5,784,964).

39. The treated claims seem to be vaguely directed towards a method of verifying the content of a downloaded applet. Gosling discloses verifying the content of computer programs written in the OAK language, later renamed Java. The Gosling invention verifies instructions of such a downloaded Java program and prevents execution of the program if errors are found. See Gosling, Summary.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jeffrey R. Swearingen whose telephone number is (571) 272-3921. The examiner can normally be reached on M-F 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Valencia Martin-Wallace can be reached on 571-272-6159. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JRS

V. Martin-Wallace

VALENCIA MARTIN-WALLACE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 3700

Form PTO- 1449 U.S. Department of Commerce Patent and Trademark Office Information Disclosure Statement by Applicant	Att. Docket Number :	Serial number : 10/069,670 RECEIVED
	Applicant : LEROY	SEP 26 2002
	Filing date : August 21, 2000	Group : Technology Center 2100

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	PATENTEE/APPLICANT	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
JM	Y	US 5,740,441	04/14/98	GOSLING JAMES A et AL			
JM		US 4,571,678	02/18/86	CHAITIN et AL			
JM		US 5,249,295	09/28/93	BRIGGS et AL			
JM		US 5,748,964	05/05/98	GOSLING JAMES A.			

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLIC. DATE	COUNTRY/AGENCY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO

Other Documents (Including Author, Title, Date, Pertinent Pages, Place of Publication, Etc.)

JM	Y	GOSLING J et AL : « The JavaTM Language Specification », JAVA LANGUAGE SPECIFICATION, XX, XX, PAGES 215-236, XP002042923, page 218, line 10-page220, line35
JM	A	GONG L et AL : « Going beyond the sandbox : an overview of the new security architecture in the JavaDevelopment Kit 1.2 ». PROCEEDINGS OF THE USENIX SYMPOSIUM ON INTERNET TECHNOLOGIES AND SYSTEMS XP002100907 Page 107, right column, line 1 – page 108, left column, line 3
JM	A	GUTHERY : " JAVA CARD : Internet Computing on a Smart Card" IEEE INTERNET COMPUTING, US, IEEE SERVICE CENTER, PICATAWAY, NJ. February 1, 1997, pages 57-59, XP002077647 ISSN : 1089 – 7801 Page 58, right column, line 19 – page 59, middle column, line 24
JM		Alfred V. AHO et AL : "Compilers : principles, techniques, and tools", Addison-Wesley 1986, Section 9.7
JM		Tim LINDHOLM, Frank YELLIN : "The java Virtual Machine Specification" Addison-Wesley 1996

JK		SUN MICROSYSTEMS INC. "JavaCard 2.1 Virtual Machine Specification", www site http://java.sun.com/products/javacard/JCVMSpec.pdf, March 1999., http://java.sun.com/produts/javacard/index.html, June 1999
JCM		GOSLING A. James : "Java Intermediate Byte Codes", proceedings of the ACM SIGPLAN, Workshop on Intermediate Representations (IR'95), pages 111-118, January 1995.
JM		ROSE et AL : "Lightweight Bytecode Verification", proceedings of the Workshop Formal Underspinning of Java, October 1998.
JP		NECULA : "Proof-Carrying Code", Proceedings of the 24th ACM Symposium Principles of Programming Languages, pages 106-119.

Examiner	Date Considered 5/3/05

EXAMINER : Initial if citation is considered, whether or not citation is in conformance with MPEP 609; Draw a line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Notice of References Cited	Application/Control No. 10/069,670	Applicant(s)/Patent Under Reexamination LEROY, XAVIER	
	Examiner Jeffrey R. Swearingen	Art Unit 2145	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,748,964	05-1998	Gosling, James A.	717/126
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier

Serial No.: 10/069,670

Filing Date: February 22, 2002

Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM

Group Art Unit: 2145

Examiner: SWEARINGEN, Jeffrey R.

Confirmation No.: 3622

Customer No.: 28465

Date of Deposit: September 29, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "First Class Mail Post Office to Addressee" service under 37 CFR § 1.8 on the date indicated above and is addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria Virginia 22313-1450.

Carole Aleman
(Typed or printed name of person mailing paper or fee)

Carole Aleman
(Signature of person mailing paper or fee)

Mail Stop Amendment
Commissioner For Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

AMENDMENT

This is in response to the Office Action dated May 16, 2005. A petition for a two month extension is enclosed. Please amend the above-identified application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims begin on page 3 of this paper.

Amendments to the Abstract begin on page 20 of this paper.

Remarks begin on page 21 of this paper.

AMENDMENTS TO THE SPECIFICATIONS

A substitute specification is attached. The substitute specification is based on a newly created translation, revised to provide an improved translation, and then amended, as indicated, to overcome other objections, as noted in the remarks section.

AMENDMENTS TO THE CLAIMS

Claims 1-3. (Canceled)

4. (Currently amended) A method of verifying a program fragment downloaded onto a reprogrammable ~~on-board~~ embedded system, ~~such as a microprocessor card~~ equipped with a rewritable memory, a microprocessor and a virtual machine equipped with an execution stack and with operand registers, said program fragment consisting of an object code and including at least one subprogram[,] consisting of a series of instructions manipulating said operand registers, ~~by the microprocessor of the on board system by way of a virtual machine equipped with an execution stack and with operand registers manipulated by these instructions, and~~ said microprocessor and virtual machine making it possible to interpret [this] said object code, said ~~on board~~ embedded system being interconnected to a reader, ~~characterized in that said method, following~~ wherein subsequent to the detection of a downloading command and the storage of said object code constituting [this] said program fragment in said rewritable memory, ~~consists,~~ said method, for each subprogram, includes:

~~a) in carrying out a stage of~~ initializing the type stack and the table of register types [by] through data representing the state of the virtual machine at the starting of the execution of [the] said temporarily stored object code;

~~b) in~~ carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching—instruction target, a target of an exception—handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

~~c) in~~ carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types[,];

~~on the basis of the existence of a branching instruction target, of a target of a subroutine call or of a target of an exception handler call~~, said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and [the] said verification process being carried out instruction by instruction until the table of register types is stable, with no modification being present, the verification process being interrupted and said program fragment being rejected, otherwise.

5. (Currently amended) The [verification]method [as claimed in]of claim 4, [characterized in that]wherein the variable types which are manipulated during [the]said verification process include at least:

[—]class identifiers corresponding to object classes which are defined in the program fragment;

[-]numeric variable types including at least a type ~~short~~, for an integer coded on [p] a given number of bits, designated as short type, and a type ~~retaddr~~ for the return address of a jump instruction [JSR], designated as a return address type;

[- a type null relating to] references of null objects designated as null type;

[— a] object type ~~object~~ relating to objects designated as object type;

[-]a first specific type [$\perp$,] representing the intersection of all the types and corresponding to the zero value [0, nil], designated as the intersection type;

[-]a second specific type [T,] representing the union of all the types and corresponding to any type of value, designated as the union type.

6. (Currently amended) The m[M]ethod ~~as claimed in~~ of claim 5, ~~characterized in that~~ wherein all said variable types verify a subtyping relation:

[object ϵ T] object type belongs to the union type;

[short, retaddr ϵ T] short type and return address type belong to the union type;

[$\perp$ ϵ null, short, retaddr] the intersection type belongs to null type, short type or return address type.

Claim 7. (Canceled)

8. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 7], [characterized in that when]wherein said current instruction [is] being the target of a subroutine call, said verification process [verifies]consists in:

verifying that the previous instruction to said current instruction is an unconditional branching, a subroutine return or a [raising]withdrawal of an exception[,]; and ~~said verification process, in the case of a positive verification, proceeding to~~ reupdat[e]ing the stack of variable types by an entity of [retaddr]the return address type, formed by the return address of the subroutine, in case of a positive verification process; and,

[the]rejecting said program fragment in case said verification process is failing, ~~and the program fragment being rejected~~ otherwise.

9. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 8], [characterized in that when the]wherein said current instruction [is] being the target of an exception handler, said verification process [verifies] consists in:

verifying that the previous instruction to said current instruction is an unconditional branching, a subroutine return or a [raising] withdrawal of an exception[,]; [said verification process, in] and

reupdating the type stack, by entering the exception type, in [the] case of a positive verification process; ~~proceeding to reupdate the type stack by entering the exception type, and the verification process failing and the program fragment being rejected~~ and

rejecting said program fragment in case of said verification process is failing, otherwise.

10. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 9], [characterized in that when the]wherein said current instruction [is]being the target of multiple incompatible branchings, [the]said verification process is fail[s]ed and [the]said program fragment is rejected.

11. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 10], [characterized in that when the]wherein said current instruction [is]being not the target of any branching, [the] said verification process [continues]consists in continuing by passing to an update of the type stack.

12. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 11], [characterized in that the stage]wherein said step of verification of the effect of the current instruction on the type stack includes, at least:

[- a stage of]verifying that the type execution stack includes at least as many entries as the current instruction includes operands;

[- a stage of]unstacking and [of] verifying that the types of the entries at the top of the stack are subtypes of the types of the operands types of the operands of [this]said current instruction;

[- a stage of]verifying the existence of a sufficient memory space on the types stack to proceed to stack the results of [the]said current instruction;

[- a stage of] stacking on the stack data types which are assigned to these results.

13. (Currently amended) The method [as claimed in]of claim 12, [characterized in that when the]wherein said current instruction [is]being an instruction to read a register of a given address [n], [the] said verification process consists in:

[— in]verifying the data type of the result of [this] a corresponding reading, by reading [the]an entry [n] at said given address in the table of register types;

[— in]determining the effect of [the]said current instruction on the type stack by unstacking the entries of the stack corresponding to the operands of [this]said current instruction and by stacking the data type of [this]said result.

14. (Currently amended) The method [as claimed in]of claim 12, [characterized in that when the] wherein said current instruction [is]being an instruction to write to a register of a given address [m], [this]said verification process consists in:

[— in]determining the effect of the current instruction on the type stack and the given type [t] of the operand which is written in this register [of]at said given address[m];

[— in]replacing the type entry of the table of register types at said given address [m]by the type immediately above the previously stored type and above the given type [t]of the operand which is written in this register [of] at said given address [m].

15. (Currently amended) A method of transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, the execution stack does not exhibit any overflow phenomenon, and for each branching instruction, the type of the stack variables at [this] a corresponding branching is the same as [at the]that of targets of this branching, into a standardized object code for this same program fragment,~~in which the operands of each instruction belong to the data types manipulated by this instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching target instruction, characterized in that this method consists,~~ wherein, for all the instructions of said object code, said method consists in:

[- in]annotating each current instruction with the data type of the stack before and after execution of [this]said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to [this]said current instruction;

[- in]detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching-targets, for which said execution stack

is not empty, [the]said detecti[on]ng operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction[,]; and in [the presence]case of detection of a non-empty execution stack,

[- in]inserting instructions to transfer stack variables on either side of [these]said branchings or of [these]said branching targets[,] respectively, in order to empty the contents of the execution stack into temporary registers before [this]said branching and to reestablish the execution stack from said temporary registers after [this]said branching[,]; and [in]not inserting any transfer instruction otherwise, [making it possible]said method allowing thus to obtain a standardized object code for [this]said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching—target instruction, in the absence of any modification to the execution of said program fragment.

16. (Currently amended) A method of transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, and an operand of given type written into a register by an instruction of this object code is reread from this same register by another instruction of [this]said object code with the same given data type, into a standardized object code for this same program fragment, ~~in which the operands of each instruction belong to the data types manipulated by this instruction, the same data type being allocated to the same register~~

~~throughout said standardized object code, characterized in that this method consists,~~wherein for all the instructions of said object code, said method consists in:

[- in]annotating each current instruction with the data type of the registers before and after execution of [this]said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to [this]said instruction;

[- in]carrying out a reallocation of [the]said registers, by detecting the original registers employed with different types, [by]dividing these original registers into separate standardized registers, with one standardized register for each data type used, and

reupdating the instructions which manipulate the operands which use said standardized registers;

said method allowing thus to obtain said standardized object code for this same program fragment in which the operands of each instruction belong to the data types manipulated by said instruction, the same data type being allocated to the same register throughout said standardized object code.

17. (Currently amended) The method [as claimed in] of claim 15, ~~characterized in that the stage consisting in~~wherein said detecting[,] within said instructions and within each current instruction[,] of the existence of branchings, or respectively of branching targets, for which the execution stack is not empty, [consists, following] after detection of each corresponding instruction of given rank [i]consists in:

[- in]associating with each instruction of said given rank [i]a set of new registers, one new register being associated with each stack variable which is active at this instruction; and

[- in]examining each detected instruction of said given rank [i]and [in] discerning the existence of a branching target or branching, respectively[,]; and, in the case where the instruction of said given rank [i] is a branching target and that the execution stack at this instruction is not empty,

[●]for every preceding instruction, of rank [i—1,]preceding said given rank and consisting of a branching, a [raising]withdrawal of an exception or a program return, [the]said detected instruction of said given rank [i] being accessible only by a branching,

[●●] [in]inserting a set of loading instructions [load]to load from the set of new registers before said detected instruction of said given rank[i], with a redirection of all branchings to the detected instruction of said given rank [i]to the first inserted loading instruction[load]; and

[●]for every preceding instruction, of rank [i—1] preceding said given rank, continuing in sequence, [the]said detected instruction of said given rank [i]being accessible simultaneously [by]from a branching and from [the]said preceding instruction of rank [i-l] preceding said given rank,

[●●] [in]inserting a set of backup instructions [store] to back up to the set of new registers before the detected instruction of said given rank[i], and a set of loading instructions [load]to load from this set of new registers, with a redirection of all

the branchings to the detected instruction of said given rank [i]to the first inserted loading instruction[load], and, in the case where said detected instruction of said given rank [i] is a branching to a given instruction,

[●]for every detected instruction of said given rank [i]consisting of an unconditional branching,

[●●][in]inserting, before the detected instruction of said given rank[i], multiple backup instructions [store], a backup instruction being associated with each new register; and

[●]for every detected instruction of said given rank [i]consisting of a conditional branching instruction, and for a given number [m > 0]greater than zero of operands manipulated by [this]said conditional branching instruction,

[●●] [in]inserting, before [this] said detected instruction of said given rank[i], a permutation instruction, [swap—x,]at the top of the execution stack of the [m]operands of the detected instruction of said given rank [i]and the [n]following values, [this]the corresponding permutation operation [making it possible]allowing thus to collect at the top of the execution stack [the n] said following values to be backed up in the set of new registers[,]; and

[●●] [in]inserting, before the instruction of said given rank[i], a set of backup instructions [store]to back up to the set of new registers[,]; and

[●●] [in]inserting, after the detected instruction of said given rank[i], a set of load instructions [load] to load from the set of new registers.

18. (Currently amended) The method [as claimed in]of claim 16, [characterized in that]wherein the [stage]step consisting in reallocating registers by detecting the original registers employed with different types consists in:

[— in]determining the lifetime intervals of each register;

[— in]determining the main data type of each lifetime interval, the main data type of a lifetime interval [j]for a given register [r]being defined by the upper bound of the data types stored in [this] said given register [r]by the backup instructions [store]belonging to [the]said lifetime interval[j];

[- in]establishing an interference graph between the lifetime intervals, [this]said interference graph consisting of a non-oriented graph of which each peak consists of a lifetime interval, and of which the arcs between two peaks [j_1 and j_2]exist if [a]one of the peaks contains a backup instruction addressed to the register of the other peak or vice versa;

[- in]translating the uniqueness of a data type which is allocated to each register in the interference graph, by adding arcs between. all pairs of peaks of the interference graph while two peaks of a pair of peaks do not have the same associated main data type;

[- in]carrying out an instantiation of the interference graph, by assigning to each lifetime interval a register number, in such a way that different register numbers are assigned to two adjacent life time intervals in [the]said interference graph.

Claim 19. (Canceled)

20. (Currently amended) An [on-board]embedded system which can be reprogrammed by downloading program fragments, said embedded system including a least one microprocessor, one random-access memory, one input/output module, one electrically reprogrammable nonvolatile memory and one permanent memory, in which are installed a main program and a virtual machine [which makes it possible]allowing to execute the main program and at least one program fragment using said microprocessor, [characterized in that]wherein said [on-board]embedded system includes at least one verification program module to [manage and]verify a downloaded program fragment in accordance with ~~the protocol for managing a downloaded program fragment as claimed in one of claims 1 to 3,~~a process including:

initializing the type stack and the table of register types through data representing the state of said virtual machine at the starting of the execution of said temporarily stored object code;

carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching-instruction target, a target of an exception-handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types;

said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and said verification process being carried out

instruction by instruction until the table of register types is stable, with no modification being present, said verification process being interrupted and said program fragment being rejected, otherwise;

said management and verification program module being installed in the permanent memory.

Claim 21. (Canceled)

22. (Currently amended) A [method of]system for transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, the execution stack does not exhibit any overflow phenomenon[,] and for each branching instruction, the type of stack variables at [this]a corresponding branching is the same as [at]that of the targets of this branching, and an operand of given type written to a register by an instruction of [this]said object code is reread from [this]said same register by another instruction of this object code with the same given data type, into a standardized object code for this same program fragment, ~~in which the operands of each instruction belong to the data types manipulated by this instruction, the execution stack does not exhibit overflow phenomenon, the execution stack is empty at each branching instruction and at each branching target instruction, the same data type being assigned to the same register throughout said standardized object code, characterized in that~~wherein said [conversion]transforming system includes, at least, installed in the working memory of a development computer or workstation, a program module [to]for transforming [this]said object code into a standardized object code in accordance with ~~the method as claimed in one of claims 15 to 18, making it possible to generate a standardized object code for said~~

~~program fragment, satisfying the criteria for verifying this downloaded program fragment~~ a process of transforming including for all the instructions of said object code:

annotating each current instruction with the data type of the stack before and after execution of said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to said current instruction;

detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching-targets, for which said execution stack is not empty, said detecting operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction; and, in case of detection of a non—empty execution stack,

inserting instructions to transfer stack variables on either side of said branchings or of said branching targets respectively, in order to empty the contents of the execution stack into temporary registers before said branching and to reestablish the execution stack from said temporary registers after said branching; and

not inserting any transfer instruction otherwise, said method allowing thus to obtain said standardized object code for said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching-target instruction, in the absence of any modification to the execution of said program fragment.

Claim 23. (Canceled)

24. (Currently amended) A computer program product which is recorded on a medium and can be loaded directly from a terminal into the internal memory of a reprogrammable [on-board] embedded system[, such as a microprocessor card] equipped with a microprocessor and a rewritable memory, [this]said [on-board]embedded system making it possible to download and temporarily store a program fragment consisting of an object code[,] including a series of instructions, executable by [the] said microprocessor [of the on-board system]by way of a virtual machine equipped with an execution stack and with operand registers manipulated via [these]said instructions and making it possible to interpret [this]said object code, [this]said computer program product including portions of object code to execute the [stages]steps of verifying a program fragment downloaded onto [this]said [on-board]embedded system ~~as claimed in one of claims 4 to 14, when this on-board system is interconnected to a terminal and this program is executed by the microprocessor of this on-board system by way of said virtual machine~~according to a verifying process, said verifying process including:

initializing the type stack and the table of register types through data representing the state of said virtual machine at the starting of the execution of said temporarily stored object code;

carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching-instruction target, a target of an exception-handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types;

said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and said verification process being carried out instruction by instruction until the table of register types is stable, with no modification being present, said verification process being interrupted and said program fragment being rejected, otherwise.

25. (Currently amended) A computer program product which is recorded on a medium including portions of object code to execute [stages]steps of [the method]a process of transforming an object code of a downloaded program fragment into a standardized object code for this same program fragment ~~as claimed in one of claims 15 to 18~~, said process of transforming including:

annotating each current instruction with the data type of the stack before and after execution of said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to said current instruction;

detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching—targets, for which said execution stack is not empty, said detecting operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction; and, in case of detection of a non-empty execution stack, inserting instructions to transfer stack variables on either side of said branchings or of said branching targets respectively, in order to empty the

contents of the execution stack into temporary registers before said branching and to reestablish the execution stack from said temporary registers after said branching; and

not inserting any transfer instruction otherwise, said method allowing thus to obtain said standardized object code for said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching—target instruction, in the absence of any modification to the execution of said program fragment.

26. (Currently amended) A computer program product which is recorded on a medium [which] and can be used in a reprogrammable [on-board]embedded system, [such as a microprocessor card]equipped with a microprocessor and a rewritable memory, [this]said [on-board]embedded system [making it possible]allowing to download a program fragment consisting of an object code, a series of instructions, executable by the microprocessor of [the]said [on-board]embedded system by [way]means of a virtual machine equipped with an execution stack and with local variables or registers manipulated via these instructions and making it possible to interpret [this]said object code, [this]said computer program product including, at least:

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a procedure for managing the downloading of a downloaded program fragment;

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a procedure for verifying, instruction by instruction, [the]said object code which makes up said program fragment;

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a downloaded program fragment [following]subsequent to or in the absence of a conversion of [the]said object code of [this]said program fragment into a standardized object code for this same program fragment.

27. (Currently amended) The computer program product as claimed in claim 26, additionally including program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command inhibition of execution, [on]by said [on-board]embedded system, of said program fragment in the case of an unsuccessful verification procedure of this program fragment.

AMENDMENTS TO THE ABSTRACT

Applicant submits an abstract set out on a separate sheet. The abstract is based on the abstract as originally published with the PCT international application. A copy of the title page of the published PCT application, providing the abstract is also attached for the Examiner's consideration.

REMARKS

SPECIFICATION

Applicant traverses the Examiner's various objections to the specification.

A substitute specification in proper idiomatic English as revised by the author of the original revised specification translation is provided. No new matter has been introduced.

The substitute specification is based on a newly created translation (translation certification enclosed), revised to provide an improved translation in proper idiomatic English. In addition, the substitute specification has been amended, as indicated in accordance with 37 CFR 1.52(a) and (b).

The present patent application is not concerned with the headings referred to as b), c), d), e) at page 3 of the Office Action.

Consequently relevant headings a), f), g), h), i), j) and k) are introduced within the specification translation to comply with the Examiner's requirements.

In particular, the following headings have been added:

BACKGROUND OF THE INVENTION

Field of the invention

Prior Art

SUMMARY OF THE INVENTION

BRIEF DESCRIPTION OF THE DRAWINGS

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embedded hyperlink references have been deleted.

Trademarks have been capitalized all along the specification translation.

With respect to minor errors in the specification translation or even clerical errors and translation errors a careful checking has been performed so as to introduce the subsequent corrections through out the specification.

Minor and clerical errors:

— Page 21 (marked up version), lines 4, 5 and 6:

The item "ε" which is obviously wrong is corrected to -∈- which clearly defines a symbol introducing the subtyping relationship of inheritance hierarchy between

classes of the applet. See particularly the subtyping relation definition at page 22 from lines 24 to 31 (marked up version).

— Page 34, lines 12, the original mentioned relationship between I_i and AI_i is clearly erroneous. It is thus corrected to read:

$I_i \leftrightarrow AI_i$

in accordance with step 500 of figure 4b, in which the same relation is introduced.

— Page 35 at line 1, the question mark ("?") is clearly erroneous. With reference to step 501 at figure 4a, in which the same relationship between AE and I_i is quoted, the query mark is corrected to $\neq$.

— Page 37, line 30, the question mark ("?") is erroneous and corrected to $\neq$ as shown at step 504a of figure 5a.

— Page 38, line 9, the character 3 is corrected to $\exists$ the existence symbol with reference to step 504 of figure 5a.

Translation errors:

Some translation errors are now corrected, through out the specification and the drawings.

— The term "on-board" is clearly wrong.

With reference to page 1 from line 16 to line 30 relating to the prior art, the dummy "on-board" data-processing systems 10 clearly refers to an -embedded- data- processing systems, as known to any person of ordinary skill in the corresponding art.

Consequently, the term "on-board" of the verified translated specification is corrected to read –embedded- through out the description.

— The term "protocol" which might appear as misleading according to the Examiner's analysis is corrected to -process- all along the description.

The title of the invention is corrected to read:

— A method for transforming and verifying downloaded program fragments with data type restrictions and corresponding system.-

An abstract of the disclosure is attached. A copy of the title page of the corresponding PCT international patent application WO 01/14958 is enclosed for the Examiner's consideration.

DRAWINGS

Applicant traverses the Examiner's objection to the drawings as failing to comply with 37 CFR 1.84 (p)(4).

Reference character 15 is objected to as having been used to designate both an EPROM and a serial link between Fig. 1a.

Referring to the substitute specification translation (marked up copy) at page 2, line 22 the correct reference number for permanent memory is "13", as also shown at Fig. la and lb.

Applicant therefore corrects reference number "15" at page 1 line 21 of the specification translation to -13-, since the objected reference number "15" clearly corresponds to a clerical error only.

Applicant respectfully traverses the Examiner's objection to the drawings under 37 CFR 1.84(p)(5). The following comments and revisions are noted.

— Reference character -13- in relation to Fig 1a is now introduced at page 1, line 21 of the specification translation.

— Reference character "103" of Fig 2 is missing in the specification translation.

The specification translation at page 16, line 19 is corrected by adding reference character -103-.

Missing reference character 103 clearly comes from a clerical error since step 103a and step 103b clearly designate a successful and an unsuccessful response respectively to corresponding verification step, that should have been labeled -103- as shown at figure 2.

— Reference character "306" of Fig 3d is missing in the specification translation.

The item -at 306- at page 25 is inserted between "stack" and "in" to read:

-the verification process reinitializes the type stack at 306 in such a way...-

— Reference character 16 of Fig 6 is missing in the specification translation.

Applicant emphasizes that figure 6 clearly refers to Fig lb.

Particularly, Fig 6 is said to disclose an embedded system 10 referred to as 10 that includes the essential components as shown at Fig lb (see particularly the specification translation at page 43 lines 15 to 18) in which item 16 is said to embody a virtual machine 16 (see particularly the specification translation at page 2 from line 24 to line 30).

— Consequently, adding further reference character 16 in relation to Fig 6 will not prove necessary, since the architecture of a reprogrammable embedded data processing system embodying a virtual machine is fully disclosed in the specification and fully know to one or ordinary skill in the corresponding art.

CLAIMS

Applicant respectfully traverses the Examiner's various objections to the claims.

Multidependency of the original claims has been canceled. The amended claims have been recasted to comply with the Examiner's remarks.

The applicant believes that in most cases the lengthy preamble is necessary to explain the invention. Wherever appropriate, the preamble has been split up, as per claims 15, 16 and 22, in which the technical features of the standardized object code which is obtained by applying the method of transforming of the invention are now recited at the end of each corresponding claim. Amending these claims this way will not introduce any new matter, since each claim content is unchanged, while the standardized object code features are now highlighted as the result which is obtained thanks to the claimed method.

Original claims 1-3 have been canceled without prejudice.

Claim 4 is amended and recast by canceling a, b, y headings and introducing corresponding indentation.

Claims 5-6 are amended by canceling the symbols $\perp$ and T. Amending claim 5 and 6 in this manner does not introduce new matter. Underlining within the claims has been omitted. Dashes to delineate steps and/or items are canceled. Bullets points within the claims are canceled.

Claims 1 to 3 are canceled and objection under 35 USC 101 directed to objected claims 1 to 3 is thus overcome.

Moreover, the specification is amended by correcting each occurrence of the item "protocol" to –process-, since, in accordance with the protocol general definition, a protocol is known to concern data exchange among given units, particularly corresponding exchange steps.

Applicant thus believes that the exchange steps better correspond to a process.

Claims 23-25 are rejected, since they are not limited to statutory computer readable media.

Claim 23 has been canceled without prejudice.

Although the computer programs which are the object of the invention are downloaded onto a reprogrammable embedded system, or a system, and thus stored therein, claims 24 and 25 are "A computer program product which is recorded on a medium." Objection under 35 USC 101 is thus overcome.

Rejection of claims 1 to 27 under 35 USC 112 first paragraph is surprising to the Applicant. Although it is agreed that the original claims correspond to a literal translation as requested by the PCT regulation requirements to enter the national phase in the United States, Applicant does not agree that the claim language is an obvious machine translation.

Some of the Examiner's objections appears unfair to the Applicant.

As an example, that "The phrase updating of the effect of said current instructions on the type stack the register table does not ensure that this takes place, thus making the claimed invention boarder than the written description" is technically and judicially unfounded and thus unfair to the Applicant.

Particularly, Applicant refers to the specification translation and corrected version at page 19 from line 25 to line 33 which contain quite the same phrase.

That the claimed invention is broader than the written description as contended by the Examiner is thus traversed.

The Examiner can either accept that the claimed and disclosed updating has taken place and the invention came to reduction to practice, or not.

In the absence of evidences given by the Examiner that reduction to practice did not take place the objection is moot or even unfair to Applicant, in case it would be maintained.

Claims 1-27 are rejected under 35 USC 112 first paragraph for the claimed invention was not described in the specification.

More particularly the virtual machine definition is not sufficient for the invention.

The Examiner's attention is drawn to the specification translation from page 1, line 11 to page 3, line 3.

Applicant believes and strongly emphasizes that the Examiner should be aware that using a virtual machine for interpreting applets within an embedded data-processing system is fully known to any person of ordinary skill in the corresponding art since 1996, as quoted with reference to the Tim LINDHOLM and Frank YELLIN publication at page 2, lines 30 to 55 of the specification translation, while the documentation edited by SUN MICROSYSTEMS Inc. on the JAVACARD 2.1 Virtual Machine Specification was available to every body since March 1999, as quoted at the paragraph spanning pages 2 and 3 of the specification translation.

That the inventor had possession of the claimed invention which is not described in such a way to reasonably convey to one skilled in the art to embody the invention is traversed.

In the absence of evidences given by the Examiner, no evidences are given that the invention unfounded to the Applicant.

Claims 1 to 27 are rejected under 35 USC 112 second paragraph for they are generally narrative and indefinite.

Amended claims are now recasted to comply with the US practice.

Claims 1-7, 15-19 and 26-27 are rejected under 35 USC 102(b) for they lack novelty over U.S. patent 5,748,964 to Gosling.

Claims 7, 19, 21 and 23 have been cancelled without prejudice.

Although the patent to Gosling is said to meet the object of the invention, Gosling does not perform the verifying method of a fragment project as the method of the invention does.

Particularly, Applicant refers to the specification translation at page 5 from lines 7 to 29 in which the mode of operation of the system as disclosed by the U.S. patent 5,748,964 to Gosling is fully acknowledged and referred to as the third solution, known from the prior art.

Applicant also refers to the International Preliminary Examination Report as established by the International Preliminary Examination Authority and the official translation thereof, of which a copy is provided herewith.

The Examiner's attention should be drawn and made aware of that the object code verifier as disclosed by Gosling, referred to as D1, has the disadvantage of a complex and costly static code verification process both in terms of the code size required to control the processor and in terms of the RAM memory size, as well as in terms of calculation time, with these memory requirements being far greater than the resource capacity of most existing embedded (on-board) computer systems.

In contradistinction to the prior art solution, the invention makes use of a method for standardizing an original object code into a standardized object code with an empty stack branch instruction using typed registers unlike the prior art methods, in which the stack type at every branching target must be stored in memory. The verification method of the invention requires only the type of the execute stack during the instruction execution being verified and does not store the stack type in memory for other subprograms. As a result, the memory capacity requirement is significantly reduced.

More particularly with reference to the substitute specification translation at page 48 line 10 to page 49 line 2, the Applicant further emphasizes that the invention is directed to a novel technique for byte code verification of JAVACARD program fragments, designated as applets, or for program fragments for similar environments.

Basically the verification operation essentially consists in requiring that:

A) the virtual operand stack be empty at each target on a branching instruction, the program fragment being thus rejected if this constraint is not satisfied;

B) the type of the local variables, designated as registers, be identical at all point within a program fragment, designated as a method, the program fragment being thus rejected if this constraint is not satisfied.

Satisfying the above mentioned constraints, in accordance with the method of the invention, allows a very efficient embedded bytecode verifier to be implemented.

As clearly quoted on the preceding highlighted paragraph of the substitute specification translation, for a given program fragment using a maximum stack size of T_p and P_r registers, the memory size required by a byte code verifier according to the invention is a direct proportion to $T_p + P_r$.

By contrast, the verifiers known from the prior art, particularly from the US patent to Gosling, would have required a memory size in a direct proportion to $(T_p + P_r) \times N_b$, the product

of $T_p + P_r$ and the number N_b of targets of branching instructions included within the program fragment.

The invention also concerns a method for transforming any program fragment accepted by any prior art verifier into a program fragment accepted by the bytecode verifier of the invention.

Consequently, while most existing or marketed program fragments or applets would possibly be rejected when submitting them to a verifier of the invention, for these existing or marketed program fragments would not necessarily satisfy the above mentioned constraints A) and B), the invention also implements the method for transforming any existing or marketed program fragment, to be verified and then executed in accordance with the method for verifying of the invention.

The invention, as implemented, appears thus fully useful for any existing or marketed program fragment and does not offend against 35 USC 101 requirements, as contended by the Examiner.

The amended claims are recasted in accordance with the preceding statement by:

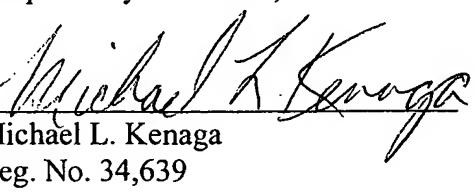
- canceling claims 1 to 3, 7, 19, 21 and 23;
- redrafting claim 4 by emphasizing corresponding constraint A) and B) which were explained before.

Remaining amended claims 4, 5, 6, 8-18, 20, 22, 24-27 are thus clearly not anticipated by Gosling and are patentable, since introducing the above discussed constraints so as to allow a significant reduction of the memory size over the most prominent bytecode verifiers of the prior art, particularly these developed by SUN MICROSYSTEMS as disclosed by Gosling, was not known or obvious at the date at which the invention was made;

- recasting claims 20, 22, 24 and 25 as independent claims to cancel reference to another claim of different category.

In view of the foregoing comments and amendment, reconsideration and allowance are requested.

Respectfully submitted,



Michael L. Kenaga
Reg. No. 34,639

DLA PIPER RUDNICK GRAY CARY US LLP
P.O. Box 64807
Chicago, Illinois 60664-0807
Phone: (312) 368-4000
Customer No.: 28465

ABSTRACT OF THE DISCLOSURE

A method and system for transforming and verifying downloaded programs fragments with data type restriction in an embedded system in which a program fragment being temporarily stored a verification process of the stored program fragment object code is executed instruction by instruction so as to discriminate for each instruction the existence of a target, a branching instruction target, a target of an exception handler call or a target of a subroutine call. On the occurrence of a target of a branching instruction as the current instruction, the empty status of the stack is verified and the program fragment is rejected otherwise. A verification process and updating of the effect of the current instruction on the data types of the type stack and the table of register types is performed. The verification process is successfully executed instruction by instruction until the table of register types is stable, with no modification being present, and interrupted with the program fragment being rejected otherwise.

TRAITE DE COOPERATION EN MATIERE DE BREVETS

Expéditeur: le BUREAU INTERNATIONAL

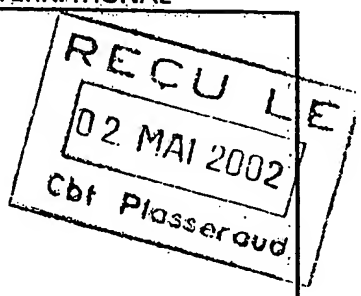
PCT

NOTIFICATION DE TRANSMISSION DE COPIES DE LA TRADUCTION DU RAPPORT D'EXAMEN PRELIMINAIRE INTERNATIONAL

(règle 72.2 du PCT)

Destinataire:

FRECHEDE, Michel
Cabinet Plasseraud
84, rue d'Amsterdam
F-75440 Paris Cedex 09
FRANCE



Date d'expédition (jour/mois/année) 16 avril 2002 (16.04.02)	
Référence du dossier du déposant ou du mandataire BCT000077	**NOTIFICATION IMPORTANTE**
Demande internationale no PCT/FR00/02349	**Date du dépôt international (jour/mois/année)** 21 août 2000 (21.08.00)
Déposant TRUSTED LOGIC etc	

1. Transmission de la traduction au déposant.

Le Bureau international transmet ci-joint copie de la traduction en langue anglaise qu'il a faite du rapport d'examen préliminaire international établi par l'administration chargée de l'examen préliminaire international.

2. Transmission d'une copie de la traduction aux offices élus.

Le Bureau international notifie au déposant qu'une copie de cette traduction a été transmise aux offices élus suivants qui exigent la traduction en question:

CN,JP,US

Les offices élus suivants ont renoncé à l'exigence selon laquelle la transmission doit être effectuée à cette date; ils recevront une copie de cette traduction du Bureau international seulement à leur demande:

EP,AU,CA

3. Rappel concernant la traduction dans la ou l'une des langues officielles de l'office ou des offices élus.

Il est rappelé au déposant que, lorsqu'une traduction de la demande internationale doit être remise à un office élu, cette traduction doit comporter la traduction de toute annexe du rapport d'examen préliminaire international.

Il appartient au déposant d'établir la traduction en question et de la remettre directement à chaque office élu intéressé (règle 74.1). Voir le volume II du Guide du déposant du PCT pour de plus amples renseignements.

Bureau international de l'OMPI
34, chemin des Colombettes
1211 Genève 20, Suisse

no de télécopieur (41-22) 740.14.35

Fonctionnaire autorisé

Maria KIRCHNER

no de téléphone (41-22) 338.83.38

Translation

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference BCT000077	FOR FURTHER ACTION	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/FR00/02349	International filing date (*day/month/year*) 21 August 2000 (21.08.00)	Priority date (*day/month/year*) 23 August 1999 (23.08.99)
International Patent Classification (IPC) or national classification and IPC G06F 9/445		
Applicant TRUSTED LOGIC		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 5 sheets, including this cover sheet.

☐ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of _____ sheets.

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☐ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☐ Lack of unity of invention
- V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☐ Certain defects in the international application
- VIII ☒ Certain observations on the international application

Date of submission of the demand	Date of completion of this report
20 March 2001 (20.03.01)	20 November 2001 (20.11.2001)
Name and mailing address of the IPEA/EP	Authorized officer
Facsimile No. —	Telephone No.

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/FR00/02349

I. Basis of the report

1. This report has been drawn on the basis of (*Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments.*):

☐ the international application as originally filed.

☒ the description, pages 1-68, as originally filed,
pages _____, filed with the demand,
pages _____, filed with the letter of _____,
pages _____, filed with the letter of _____.

☒ the claims, Nos. 1-27, as originally filed,
Nos. _____, as amended under Article 19,
Nos. _____, filed with the demand,
Nos. _____, filed with the letter of _____,
Nos. _____, filed with the letter of _____.

☒ the drawings, sheets/fig 1-14, as originally filed,
sheets/fig _____, filed with the demand,
sheets/fig _____, filed with the letter of _____,
sheets/fig _____, filed with the letter of _____.

2. The amendments have resulted in the cancellation of:

☐ the description, pages _____
☐ the claims, Nos. _____
☐ the drawings, sheets/fig _____

3. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).

4. Additional observations, if necessary:

V. **Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-27	YES
	Claims		NO
Inventive step (IS)	Claims	1-27	YES
	Claims		NO
Industrial applicability (IA)	Claims	1-27	YES
	Claims		NO

2. Citations and explanations

1. Reference is made to the following document:

D1: US-A-5 748 964

2. The present invention relates to a method for converting a conventional object code constituting an applet for execution by an on-board computer system having limited resources.

Prior art:

The download of an applet to an on-board computer system is subject to authenticity verification. D1 discloses static verification simulating the execution of the applet with data types and ensures, once and for all, that the applet code complies with the data type and access control rules set by the virtual machine and that it does not cause stack overflow.

Problem:

This solution has the disadvantage of a complex and costly static code verification process both in

terms of the code size required to control the processor and in terms of the RAM memory size required to contain the intermediate verification results, as well as in terms of the calculation time. These memory requirements are far greater than the resource capacity of most existing on-board computer systems.

Solution:

The present invention uses a process for standardising an original object code into a standardised object code with an empty stack branch instruction and a standardised code using typed registers such that any one register is used under a single type throughout the subprogram code. Unlike the prior art methods, in which the stack type at every branch target must be stored in memory, the verification method of the present invention requires only the type of the execute stack during the instruction being verified and does not store the stack type in memory for other subprograms. As a result, the memory capacity requirement is reduced.

3. The dependent claims relate to specific embodiments of the invention according to the independent claims. Therefore, they too comply with the requirements of novelty, inventive step and industrial applicability.

VIII. Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made:

It is clear from page 38 of the description that the feature whereby the stack must be empty for each branch or branch target instruction, and/or all of the registers are reset when the method is initialised, is essential for the definition of the invention. Indeed, the phrase "updating of the effect of said current instruction on the type stack and the register type table" in no way means that conditions C3 and C4 (page 38), which are fundamental to the present invention, have been met.

Since the independent claims do not contain these features, they fail to comply with the requirements of PCT Article 6 in combination with PCT Rule 6.3(b), according to which an independent claim must contain all of the technical features essential for the definition of the invention. Therefore, the independent claims do not meet the requirements of PCT Article 6.

TRANSLATION CERTIFICATION

RWS Group Ltd, of Europa House, Marsham Way, Gerrards Cross, Buckinghamshire, England, hereby declares that, to the best of its knowledge and belief, the following document, prepared by one of its translators competent in the art and conversant with the English and French languages, is a true and correct translation of the accompanying document in the French language.

Signed this 4th day of August 2005

C. E. SITCH

Deputy Managing Director - UK Translation Division

For and on behalf of RWS Group Ltd

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
1 mars 2001 (01.03.2001)

PCT

(10) Numéro de publication internationale
WO 01/14958 A2

(51) Classification internationale des brevets[7]: G06F 9/00

(21) Numéro de la demande internationale:
PCT/FR00/02349

(22) Date de dépôt international: 21 août 2000 (21.08.2000)

(25) Langue de dépôt: français

(26) Langue de publication: français

(30) Données relatives à la priorité:
99/10697 23 août 1999 (23.08.1999) FR

(71) Déposant *(pour tous les États désignés sauf US)*: **TRUSTED LOGIC** [FR/FR]; 23, avenue de Fulpmes, F-78450 Villepreux (FR).

(72) Inventeur; et

(75) Inventeur/Déposant *(pour US seulement)*: **LEROY, Xavier** [FR/FR]; 88 bis, avenue de Paris, F-78000 Versailles (FR).

(74) Mandataires: **FRECHEDE, Michel** etc.; Cabinet Plasseraud, 84, rue d'Amsterdam, F-75440 Paris Cedex 09 (FR).

(81) États désignés *(national)*: AU, CA, CN, JP, US.

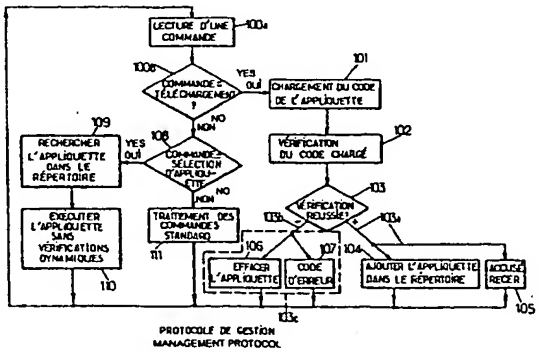
(84) États désignés *(régional)*: brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Publiée:
— *Sans rapport de recherche internationale, sera republiée dès réception de ce rapport.*

[Suite sur la page suivante]

(54) Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS

(54) Titre: PROTOCOLE DE GESTION, PROCEDE DE VERIFICATION ET DE TRANSFORMATION D'UN FRAGMENT DE PROGRAMME TELECHARGE ET SYSTEMES CORRESPONDANTS



(57) Abstract: The invention relates to a management protocol and to a method for verifying a programme fragment, or applet, which has been downloaded onto a portable system. An applet downloading command (100a, 100b) is executed. Once a positive response has been received, the object code of the applet is read (101) and subjected (102) to a verification process, instruction by instruction. The verification process consists of a stage comprising the initialisation of the type stack and table of register types representing the state of the virtual machine of the portable system at the start of the execution of the applet code; and a verification, instruction by instruction, for each target current instruction, of the existence of a target branch instruction, a target exception handler call or a target sub-routine call, the effect of the instruction on the type stack and the table of register types being verified and updated. If the verification is successful (103a), the applet is registered (104) and an acknowledgement is sent (105) to the downloading drive. Otherwise, the applet is destroyed (106). The invention is suitable for use for portable systems in a Java environment.

[Suite sur la page suivante]

WO 01/14958 A2

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) (Large Entity)	Docket No. P6451 (218728-000123)

In Re Application Of: **Xavier LEROY**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
10/069,670	Feb. 22, 2002	Jeffrey R. Swearingen	28465	2145	3622

Invention: **A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM**

COMMISSIONER FOR PATENTS:

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a response to the Office Action of May 16, 2005 (*Date*) above-identified application.

The requested extension is as follows (check time period desired):

☐ One month ☒ Two months ☐ Three months ☐ Four months ☐ Five months

from: August 16, 2005 (*Date*) until: October 16, 2005 (*Date*)

The fee for the extension of time is $450 and is to be paid as follows:

☒ A check in the amount of the fee is enclosed.

☐ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account No.

☒ If an additional extension of time is required, please consider this a petition therefor and charge any additional fees which may be required to Deposit Account No. 18-2284

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Michael L Kenaga
Signature

Dated: September 29, 2005

Michael L. Kenaga
Registration No. 34,639

DLA PIPER RUDNICK GRAY CARY US LLP
P.O. Box 64807
Chicago, Illinois 60664-0807
(312) 368-4000

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on September 29, 2005 (Date).

Carole Aleman
Signature of Person Mailing Correspondence

Carole Aleman
Typed or Printed Name of Person Mailing Correspondence

cc:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier
Serial No.: 10/069,670; Filed: February 22, 2002
Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM
Group Art Unit: 2145
Examiner: SWEARINGEN, Jeffrey R.
Conf. No.: 3622
Customer No.: 28465
Our Ref.: P6451 (218728-000123)

Enclosures:

1. Amendment, including Abstract sheet attached thereto (31 pages)
2. International Preliminary Examination Report for PCT/FR00/02349, including an official translation thereof (6 pages)
3. Translation Certification (2 pages)
4. Title Page of Published PCT Application No. PCT/FR00/02349
5. Petition for Extension of Time Under 37 CFR 1.136(a) (1 page) and duplicate thereof
6. Check No. 074467 in the amount of $450.00
7. Return receipt postcard

Mailed via First Class Mail on September 28, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier
Serial No.: 10/069,670; Filed: February 22, 2002
Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM
Group Art Unit: 2145
Examiner: SWEARINGEN, Jeffrey R.
Conf. No.: 3622
Customer No.: 28465
Our Ref.: P6451 (218728-000123)

Enclosures:

1. Amendment, including Abstract sheet attached thereto (31 pages)
2. International Preliminary Examination Report for PCT/FR00/02349, including an official translation thereof (6 pages)
3. Translation Certification (2 pages)
4. Title Page of Published PCT Application No. PCT/FR00/02349
5. Petition for Extension of Time Under 37 CFR 1.136(a) (1 page) and duplicate thereof
6. Check No. 074467 in the amount of $450.00
7. Return receipt postcard

Mailed via First Class Mail on September 28, 2005

Jul.

Attorney Docket No: P6451

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier

Serial No.: 10/069,670

Filing Date: February 22, 2002

Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM

Group Art Unit: 2145

Examiner: SWEARINGEN, Jeffrey R.

Confirmation No.: 3622

Customer No.: 28465

Date of Deposit: October 31, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "First Class Mail Post Office to Addressee" service under 37 CFR § 1.8 on the date indicated above and is addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria Virginia 22313-1450.

Carole Aleman
(Typed or printed name of person mailing paper or fee)

Carole Aleman
(Signature of person mailing paper or fee)

Mail Stop Amendment
Commissioner For Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

SUPPLEMENTAL AMENDMENT

This is in response to the Office Action dated May 16, 2005. Please amend the above-identified application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims begin on page 3 of this paper.

Amendments to the Abstract begin on page 20 of this paper.

Remarks begin on page 21 of this paper.

AMENDMENTS TO THE SPECIFICATIONS

A substitute specification is attached. The substitute specification is based on a newly created translation, revised to provide an improved translation, and then amended, as indicated, to overcome other objections, as noted in the remarks section.

AMENDMENTS TO THE CLAIMS

Claims 1-3. (Canceled)

4. (Currently amended) A method of verifying a program fragment downloaded onto a reprogrammable ~~on-board~~ embedded system, ~~such as a microprocessor card~~ equipped with a rewritable memory, a microprocessor and a virtual machine equipped with an execution stack and with operand registers, said program fragment consisting of an object code and including at least one subprogram[,] consisting of a series of instructions manipulating said operand registers, ~~by the microprocessor of the on board system by way of a virtual machine equipped with an execution stack and with operand registers manipulated by these instructions, and~~ said microprocessor and virtual machine making it possible to interpret [this] said object code, said ~~on board~~ embedded system being interconnected to a reader, ~~characterized in that said method, following~~ wherein subsequent to the detection of a downloading command and the storage of said object code constituting [this] said program fragment in said rewritable memory, ~~consists,~~ said method, for each subprogram, includes:

~~a) in carrying out a stage of~~ initializing the type stack and the table of register types [by] through data representing the state of the virtual machine at the starting of the execution of [the] said temporarily stored object code;

~~b) in~~ carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching—instruction target, a target of an exception—handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

~~c) in~~ carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types[,];

~~on the basis of the existence of a branching instruction target, of a target of a subroutine call or of a target of an exception handler call~~, said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and [the] said verification process being carried out instruction by instruction until the table of register types is stable, with no modification being present, the verification process being interrupted and said program fragment being rejected, otherwise.

5. (Currently amended) The [verification]method [as claimed in]of claim 4, [characterized in that]wherein the variable types which are manipulated during [the]said verification process include at least:

[—]class identifiers corresponding to object classes which are defined in the program fragment;

[-]numeric variable types including at least a type ~~short,~~ for an integer coded on [p] a given number of bits, designated as short type, and a type ~~retaddr~~ for the return address of a jump instruction [JSR], designated as a return address type;

[- a type null relating to] references of null objects designated as null type;

[— a] object type ~~object~~ relating to objects designated as object type;

[-]a first specific type [⊥,] representing the intersection of all the types and corresponding to the zero value [0, nil], designated as the intersection type;

[-]a second specific type [T,] representing the union of all the types and corresponding to any type of value, designated as the union type.

6. (Currently amended) The m[M]ethod ~~as claimed in~~ of claim 5, ~~characterized in that~~ wherein all said variable types verify a subtyping relation:

[object ϵ T] object type belongs to the union type;

[short, retaddr ϵ T] short type and return address type belong to the union type;

[$\perp$ ϵ null, short, retaddr] the intersection type belongs to null type, short type or return address type.

Claim 7. (Canceled)

8. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 7], [characterized in that when]wherein said current instruction [is] being the target of a subroutine call, said verification process [verifies]consists in:

verifying that the previous instruction to said current instruction is an unconditional branching, a subroutine return or a [raising]withdrawal of an exception[,]; and ~~said verification process, in the case of a positive verification, proceeding to~~ reupdat[e]ing the stack of variable types by an entity of [retaddr]the return address type, formed by the return address of the subroutine, in case of a positive verification process; and,

[the]rejecting said program fragment in case said verification process is failing, ~~and the program fragment being rejected~~ otherwise.

9. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 8], [characterized in that when the]wherein said current instruction [is] being the target of an exception handler, said verification process [verifies] consists in:

verifying that the previous instruction to said current instruction is an unconditional branching, a subroutine return or a [raising] withdrawal of an exception[,]; [said verification process, in] and

reupdating the type stack, by entering the exception type, in [the] case of a positive verification process; ~~proceeding to reupdate the type stack by entering the exception type, and the verification process failing and the program fragment being rejected~~ and

rejecting said program fragment in case of said verification process is failing, otherwise.

10. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 9], [characterized in that when the]wherein said current instruction [is]being the target of multiple incompatible branchings, [the]said verification process is fail[s]ed and [the]said program fragment is rejected.

11. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 10], [characterized in that when the]wherein said current instruction [is]being not the target of any branching, [the] said verification process [continues]consists in continuing by passing to an update of the type stack.

12. (Currently amended) The method [as claimed in one] of claim[s] 4 [to 11], [characterized in that the stage]wherein said step of verification of the effect of the current instruction on the type stack includes, at least:

[- a stage of]verifying that the type execution stack includes at least as many entries as the current instruction includes operands;

[- a stage of]unstacking and [of] verifying that the types of the entries at the top of the stack are subtypes of the types of the operands types of the operands of [this]said current instruction;

[- a stage of]verifying the existence of a sufficient memory space on the types stack to proceed to stack the results of [the]said current instruction;

[- a stage of] stacking on the stack data types which are assigned to these results.

13. (Currently amended) The method [as claimed in]of claim 12, [characterized in that when the]wherein said current instruction [is]being an instruction to read a register of a given address [n], [the] said verification process consists in:

[— in]verifying the data type of the result of [this] a corresponding reading, by reading [the]an entry [n] at said given address in the table of register types;

[— in]determining the effect of [the]said current instruction on the type stack by unstacking the entries of the stack corresponding to the operands of [this]said current instruction and by stacking the data type of [this]said result.

14. (Currently amended) The method [as claimed in]of claim 12, [characterized in that when the] wherein said current instruction [is]being an instruction to write to a register of a given address [m], [this]said verification process consists in:

[— in]determining the effect of the current instruction on the type stack and the given type [t] of the operand which is written in this register [of]at said given address[m];

[— in]replacing the type entry of the table of register types at said given address [m]by the type immediately above the previously stored type and above the given type [t]of the operand which is written in this register [of] at said given address [m].

15. (Currently amended) A method of transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, the execution stack does not exhibit any overflow phenomenon, and for each branching instruction, the type of the stack variables at [this] a corresponding branching is the same as [at the]that of targets of this branching, into a standardized object code for this same program fragment,~~in which the operands of each instruction belong to the data types manipulated by this instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching target instruction, characterized in that this method consists,~~ wherein, for all the instructions of said object code, said method consists in:

[- in]annotating each current instruction with the data type of the stack before and after execution of [this]said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to [this]said current instruction;

[- in]detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching-targets, for which said execution stack

is not empty, [the]said detecti[on]ng operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction[,]; and in [the presence]case of detection of a non-empty execution stack,

[- in]inserting instructions to transfer stack variables on either side of [these]said branchings or of [these]said branching targets[,] respectively, in order to empty the contents of the execution stack into temporary registers before [this]said branching and to reestablish the execution stack from said temporary registers after [this]said branching[,]; and [in]not inserting any transfer instruction otherwise, [making it possible]said method allowing thus to obtain a standardized object code for [this]said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching—target instruction, in the absence of any modification to the execution of said program fragment.

16. (Currently amended) A method of transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, and an operand of given type written into a register by an instruction of this object code is reread from this same register by another instruction of [this]said object code with the same given data type, into a standardized object code for this same program fragment, ~~in which the operands of each instruction belong to the data types manipulated by this instruction, the same data type being allocated to the same register~~

~~throughout said standardized object code, characterized in that this method consists,~~wherein for all the instructions of said object code, said method consists in:

[- in]annotating each current instruction with the data type of the registers before and after execution of [this]said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to [this]said instruction;

[- in]carrying out a reallocation of [the]said registers, by detecting the original registers employed with different types, [by]dividing these original registers into separate standardized registers, with one standardized register for each data type used, and

reupdating the instructions which manipulate the operands which use said standardized registers;

said method allowing thus to obtain said standardized object code for this same program fragment in which the operands of each instruction belong to the data types manipulated by said instruction, the same data type being allocated to the same register throughout said standardized object code.

17. (Currently amended) The method [as claimed in] of claim 15, ~~characterized in that the stage consisting in~~wherein said detecting[,] within said instructions and within each current instruction[,] of the existence of branchings, or respectively of branching targets, for which the execution stack is not empty, [consists, following] after detection of each corresponding instruction of given rank [i]consists in:

[- in]associating with each instruction of said given rank [i]a set of new registers, one new register being associated with each stack variable which is active at this instruction; and

[- in]examining each detected instruction of said given rank [i]and [in] discerning the existence of a branching target or branching, respectively[,]; and, in the case where the instruction of said given rank [i] is a branching target and that the execution stack at this instruction is not empty,

[●]for every preceding instruction, of rank [i—1,]preceding said given rank and consisting of a branching, a [raising]withdrawal of an exception or a program return, [the]said detected instruction of said given rank [i] being accessible only by a branching,

[●●] [in]inserting a set of loading instructions [load]to load from the set of new registers before said detected instruction of said given rank[i], with a redirection of all branchings to the detected instruction of said given rank [i]to the first inserted loading instruction[load]; and

[●]for every preceding instruction, of rank [i—1] preceding said given rank, continuing in sequence, [the]said detected instruction of said given rank [i]being accessible simultaneously [by]from a branching and from [the]said preceding instruction of rank [i-l] preceding said given rank,

[●●] [in]inserting a set of backup instructions [store] to back up to the set of new registers before the detected instruction of said given rank[i], and a set of loading instructions [load]to load from this set of new registers, with a redirection of all

the branchings to the detected instruction of said given rank [i]to the first inserted loading instruction[load], and, in the case where said detected instruction of said given rank [i] is a branching to a given instruction,

[●]for every detected instruction of said given rank [i]consisting of an unconditional branching,

[●●][in]inserting, before the detected instruction of said given rank[i], multiple backup instructions [store], a backup instruction being associated with each new register; and

[●]for every detected instruction of said given rank [i]consisting of a conditional branching instruction, and for a given number [m > 0]greater than zero of operands manipulated by [this]said conditional branching instruction,

[●●] [in]inserting, before [this] said detected instruction of said given rank[i], a permutation instruction, [swap—x,]at the top of the execution stack of the [m]operands of the detected instruction of said given rank [i]and the [n]following values, [this]the corresponding permutation operation [making it possible]allowing thus to collect at the top of the execution stack [the n] said following values to be backed up in the set of new registers[,]; and

[●●] [in]inserting, before the instruction of said given rank[i], a set of backup instructions [store]to back up to the set of new registers[,]; and

[●●] [in]inserting, after the detected instruction of said given rank[i], a set of load instructions [load] to load from the set of new registers.

18. (Currently amended) The method [as claimed in]of claim 16, [characterized in that]wherein the [stage]step consisting in reallocating registers by detecting the original registers employed with different types consists in:

[— in]determining the lifetime intervals of each register;

[— in]determining the main data type of each lifetime interval, the main data type of a lifetime interval [j]for a given register [r]being defined by the upper bound of the data types stored in [this] said given register [r]by the backup instructions [store]belonging to [the]said lifetime interval[j];

[- in]establishing an interference graph between the lifetime intervals, [this]said interference graph consisting of a non-oriented graph of which each peak consists of a lifetime interval, and of which the arcs between two peaks [j_1 and j_2]exist if [a]one of the peaks contains a backup instruction addressed to the register of the other peak or vice versa;

[- in]translating the uniqueness of a data type which is allocated to each register in the interference graph, by adding arcs between. all pairs of peaks of the interference graph while two peaks of a pair of peaks do not have the same associated main data type;

[- in]carrying out an instantiation of the interference graph, by assigning to each lifetime interval a register number, in such a way that different register numbers are assigned to two adjacent life time intervals in [the]said interference graph.

Claim 19. (Canceled)

20. (Currently amended) An [on-board]embedded system which can be reprogrammed by downloading program fragments, said embedded system including a least one microprocessor, one random-access memory, one input/output module, one electrically reprogrammable nonvolatile memory and one permanent memory, in which are installed a main program and a virtual machine [which makes it possible]allowing to execute the main program and at least one program fragment using said microprocessor, [characterized in that]wherein said [on-board]embedded system includes at least one verification program module to [manage and]verify a downloaded program fragment in accordance with ~~the protocol for managing a downloaded program fragment as claimed in one of claims 1 to 3,~~a process including:

initializing the type stack and the table of register types through data representing the state of said virtual machine at the starting of the execution of said temporarily stored object code;

carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching-instruction target, a target of an exception-handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types;

said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and said verification process being carried out

instruction by instruction until the table of register types is stable, with no modification being present, said verification process being interrupted and said program fragment being rejected, otherwise;

said management and verification program module being installed in the permanent memory.

Claim 21. (Canceled)

22. (Currently amended) A [method of]system for transforming an object code of a program fragment including a series of instructions, in which the operands of each instruction belong to the data types manipulated by [this]said instruction, the execution stack does not exhibit any overflow phenomenon[,] and for each branching instruction, the type of stack variables at [this]a corresponding branching is the same as [at]that of the targets of this branching, and an operand of given type written to a register by an instruction of [this]said object code is reread from [this]said same register by another instruction of this object code with the same given data type, into a standardized object code for this same program fragment, ~~in which the operands of each instruction belong to the data types manipulated by this instruction, the execution stack does not exhibit overflow phenomenon, the execution stack is empty at each branching instruction and at each branching target instruction, the same data type being assigned to the same register throughout said standardized object code, characterized in that~~wherein said [conversion]transforming system includes, at least, installed in the working memory of a development computer or workstation, a program module [to]for transforming [this]said object code into a standardized object code in accordance with ~~the method as claimed in one of claims 15 to 18, making it possible to generate a standardized object code for said~~

~~program fragment, satisfying the criteria for verifying this downloaded program fragment~~ a process of transforming including for all the instructions of said object code:

annotating each current instruction with the data type of the stack before and after execution of said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to said current instruction;

detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching-targets, for which said execution stack is not empty, said detecting operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction; and, in case of detection of a non—empty execution stack,

inserting instructions to transfer stack variables on either side of said branchings or of said branching targets respectively, in order to empty the contents of the execution stack into temporary registers before said branching and to reestablish the execution stack from said temporary registers after said branching; and

not inserting any transfer instruction otherwise, said method allowing thus to obtain said standardized object code for said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching-target instruction, in the absence of any modification to the execution of said program fragment.

Claim 23. (Canceled)

24. (Currently amended) A computer program product which is recorded on a medium and can be loaded directly from a terminal into the internal memory of a reprogrammable [on-board] embedded system[, such as a microprocessor card] equipped with a microprocessor and a rewritable memory, [this]said [on-board]embedded system making it possible to download and temporarily store a program fragment consisting of an object code[,] including a series of instructions, executable by [the] said microprocessor [of the on-board system]by way of a virtual machine equipped with an execution stack and with operand registers manipulated via [these]said instructions and making it possible to interpret [this]said object code, [this]said computer program product including portions of object code to execute the [stages]steps of verifying a program fragment downloaded onto [this]said [on-board]embedded system ~~as claimed in one of claims 4 to 14, when this on-board system is interconnected to a terminal and this program is executed by the microprocessor of this on-board system by way of said virtual machine~~according to a verifying process, said verifying process including:

initializing the type stack and the table of register types through data representing the state of said virtual machine at the starting of the execution of said temporarily stored object code;

carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning the existence, for each current instruction, of a target, a branching-instruction target, a target of an exception-handler call or a target of a subroutine call, and, said current instruction being the target of a branching instruction, said verification process consisting in verifying that the stack is empty and rejecting the program fragment otherwise;

carrying out a verification process and an updating of the effect of said current instruction on the data types of said type stack and of said table of register types;

said verification process being successful when the table of register types is not modified in the course of a verification of all the instructions, and said verification process being carried out instruction by instruction until the table of register types is stable, with no modification being present, said verification process being interrupted and said program fragment being rejected, otherwise.

25. (Currently amended) A computer program product which is recorded on a medium including portions of object code to execute [stages]steps of [the method]a process of transforming an object code of a downloaded program fragment into a standardized object code for this same program fragment ~~as claimed in one of claims 15 to 18~~, said process of transforming including:

annotating each current instruction with the data type of the stack before and after execution of said current instruction, with the annotation data being calculated by means of an analysis of the data stream relating to said current instruction;

detecting, within said instructions and within each current instruction, the existence of branchings, or respectively of branching—targets, for which said execution stack is not empty, said detecting operation being carried out on the basis of the annotation data of the type of stack variables allocated to each current instruction, and, in case of detection of a non-empty execution stack;

inserting instructions to transfer stack variables on either side of said branchings or of said branching targets respectively, in order to empty the contents of the execution stack into

temporary registers before said branching and to reestablish the execution stack from said temporary registers after said branching; and

not inserting any transfer instruction otherwise, said method allowing thus to obtain said standardized object code for said same program fragment, in which the operands of each instruction belong to the data types manipulated by said instruction, the execution stack does not exhibit any overflow phenomenon, the execution stack is empty at each branching instruction and at each branching—target instruction, in the absence of any modification to the execution of said program fragment.

26. (Currently amended) A computer program product which is recorded on a medium [which] and can be used in a reprogrammable [on-board]embedded system, [such as a microprocessor card]equipped with a microprocessor and a rewritable memory, [this]said [on-board]embedded system [making it possible]allowing to download a program fragment consisting of an object code, a series of instructions, executable by the microprocessor of [the]said [on-board]embedded system by [way]means of a virtual machine equipped with an execution stack and with local variables or registers manipulated via these instructions and making it possible to interpret [this]said object code, [this]said computer program product including, at least:

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a procedure for managing the downloading of a downloaded program fragment;

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a procedure for verifying, instruction by instruction, [the]said object code which makes up said program fragment;

[-]program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command execution of a downloaded program fragment [following]subsequent to or in the absence of a conversion of [the]said object code of [this]said program fragment into a standardized object code for this same program fragment.

27. (Currently amended) The computer program product as claimed in claim 26, additionally including program resources which can be read by the microprocessor of [this]said [on-board]embedded system via said virtual machine, to command inhibition of execution, [on]by said [on-board]embedded system, of said program fragment in the case of an unsuccessful verification procedure of this program fragment.

AMENDMENTS TO THE ABSTRACT

Applicant submits an abstract set out on a separate sheet. The abstract is based on the abstract as originally published with the PCT international application. A copy of the title page of the published PCT application, providing the abstract is also attached for the Examiner's consideration.

REMARKS

SPECIFICATION

Applicant traverses the Examiner's various objections to the specification.

A substitute specification in proper idiomatic English as revised by the author of the original revised specification translation is provided. No new matter has been introduced.

The substitute specification is based on a newly created translation (translation certification enclosed), revised to provide an improved translation in proper idiomatic English. In addition, the substitute specification has been amended, as indicated in accordance with 37 CFR 1.52(a) and (b).

The present patent application is not concerned with the headings referred to as b), c), d), e) at page 3 of the Office Action.

Consequently relevant headings a), f), g), h), i), j) and k) are introduced within the specification translation to comply with the Examiner's requirements.

In particular, the following headings have been added:

BACKGROUND OF THE INVENTION

Field of the invention

Prior Art

SUMMARY OF THE INVENTION

BRIEF DESCRIPTION OF THE DRAWINGS

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embedded hyperlink references have been deleted.

Trademarks have been capitalized all along the specification translation.

With respect to minor errors in the specification translation or even clerical errors and translation errors a careful checking has been performed so as to introduce the subsequent corrections through out the specification.

Minor and clerical errors:

— Page 21 (marked up version), lines 4, 5 and 6:

The item "ε" which is obviously wrong is corrected to -∈- which clearly defines a symbol introducing the subtyping relationship of inheritance hierarchy between

classes of the applet. See particularly the subtyping relation definition at page 22 from lines 24 to 31 (marked up version).

— Page 34, lines 12, the original mentioned relationship between I_i and AI_i is clearly erroneous. It is thus corrected to read:

$I_i \leftrightarrow AI_i$

in accordance with step 500 of figure 4b, in which the same relation is introduced.

— Page 35 at line 1, the question mark ("?") is clearly erroneous. With reference to step 501 at figure 4a, in which the same relationship between AE and I_i is quoted, the query mark is corrected to $\neq$.

— Page 37, line 30, the question mark ("?") is erroneous and corrected to $\neq$ as shown at step 504a of figure 5a.

— Page 38, line 9, the character 3 is corrected to $\exists$ the existence symbol with reference to step 504 of figure 5a.

Translation errors:

Some translation errors are now corrected, through out the specification and the drawings.

— The term "on-board" is clearly wrong.

With reference to page 1 from line 16 to line 30 relating to the prior art, the dummy "on-board" data-processing systems 10 clearly refers to an -embedded- data- processing systems, as known to any person of ordinary skill in the corresponding art.

Consequently, the term "on-board" of the verified translated specification is corrected to read –embedded- through out the description.

— The term "protocol" which might appear as misleading according to the Examiner's analysis is corrected to -process- all along the description.

The title of the invention is corrected to read:

— A method for transforming and verifying downloaded program fragments with data type restrictions and corresponding system.-

An abstract of the disclosure is attached. A copy of the title page of the corresponding PCT international patent application WO 01/14958 is enclosed for the Examiner's consideration.

DRAWINGS

Applicant traverses the Examiner's objection to the drawings as failing to comply with 37 CFR 1.84 (p)(4).

Reference character 15 is objected to as having been used to designate both an EPROM and a serial link between Fig. 1a.

Referring to the substitute specification translation (marked up copy) at page 2, line 22 the correct reference number for permanent memory is "13", as also shown at Fig. la and lb.

Applicant therefore corrects reference number "15" at page 1 line 21 of the specification translation to -13-, since the objected reference number "15" clearly corresponds to a clerical error only.

Applicant respectfully traverses the Examiner's objection to the drawings under 37 CFR 1.84(p)(5). The following comments and revisions are noted.

— Reference character -13- in relation to Fig 1a is now introduced at page 1, line 21 of the specification translation.

— Reference character "103" of Fig 2 is missing in the specification translation.

The specification translation at page 16, line 19 is corrected by adding reference character -103-.

Missing reference character 103 clearly comes from a clerical error since step 103a and step 103b clearly designate a successful and an unsuccessful response respectively to corresponding verification step, that should have been labeled -103- as shown at figure 2.

— Reference character "306" of Fig 3d is missing in the specification translation.

The item -at 306- at page 25 is inserted between "stack" and "in" to read:

-the verification process reinitializes the type stack at 306 in such a way...-

— Reference character 16 of Fig 6 is missing in the specification translation.

Applicant emphasizes that figure 6 clearly refers to Fig lb.

Particularly, Fig 6 is said to disclose an embedded system 10 referred to as 10 that includes the essential components as shown at Fig lb (see particularly the specification translation at page 43 lines 15 to 18) in which item 16 is said to embody a virtual machine 16 (see particularly the specification translation at page 2 from line 24 to line 30).

— Consequently, adding further reference character 16 in relation to Fig 6 will not prove necessary, since the architecture of a reprogrammable embedded data processing system embodying a virtual machine is fully disclosed in the specification and fully know to one or ordinary skill in the corresponding art.

CLAIMS

Applicant respectfully traverses the Examiner's various objections to the claims.

Multidependency of the original claims has been canceled. The amended claims have been recasted to comply with the Examiner's remarks.

The applicant believes that in most cases the lengthy preamble is necessary to explain the invention. Wherever appropriate, the preamble has been split up, as per claims 15, 16 and 22, in which the technical features of the standardized object code which is obtained by applying the method of transforming of the invention are now recited at the end of each corresponding claim. Amending these claims this way will not introduce any new matter, since each claim content is unchanged, while the standardized object code features are now highlighted as the result which is obtained thanks to the claimed method.

Original claims 1-3 have been canceled without prejudice.

Claim 4 is amended and recast by canceling a, b, y headings and introducing corresponding indentation.

Claims 5-6 are amended by canceling the symbols $\perp$ and T. Amending claim 5 and 6 in this manner does not introduce new matter. Underlining within the claims has been omitted. Dashes to delineate steps and/or items are canceled. Bullets points within the claims are canceled.

Claims 1 to 3 are canceled and objection under 35 USC 101 directed to objected claims 1 to 3 is thus overcome.

Moreover, the specification is amended by correcting each occurrence of the item "protocol" to –process-, since, in accordance with the protocol general definition, a protocol is known to concern data exchange among given units, particularly corresponding exchange steps.

Applicant thus believes that the exchange steps better correspond to a process.

Claims 23-25 are rejected, since they are not limited to statutory computer readable media.

Claim 23 has been canceled without prejudice.

Although the computer programs which are the object of the invention are downloaded onto a reprogrammable embedded system, or a system, and thus stored therein, claims 24 and 25 are "A computer program product which is recorded on a medium." Objection under 35 USC 101 is thus overcome.

Rejection of claims 1 to 27 under 35 USC 112 first paragraph is surprising to the Applicant. Although it is agreed that the original claims correspond to a literal translation as requested by the PCT regulation requirements to enter the national phase in the United States, Applicant does not agree that the claim language is an obvious machine translation.

Some of the Examiner's objections appears unfair to the Applicant.

As an example, that "The phrase updating of the effect of said current instructions on the type stack the register table does not ensure that this takes place, thus making the claimed invention boarder than the written description" is technically and judicially unfounded and thus unfair to the Applicant.

Particularly, Applicant refers to the specification translation and corrected version at page 19 from line 25 to line 33 which contain quite the same phrase.

That the claimed invention is broader than the written description as contended by the Examiner is thus traversed.

The Examiner can either accept that the claimed and disclosed updating has taken place and the invention came to reduction to practice, or not.

In the absence of evidences given by the Examiner that reduction to practice did not take place the objection is moot or even unfair to Applicant, in case it would be maintained.

Claims 1-27 are rejected under 35 USC 112 first paragraph for the claimed invention was not described in the specification.

More particularly the virtual machine definition is not sufficient for the invention.

The Examiner's attention is drawn to the specification translation from page 1, line 11 to page 3, line 3.

Applicant believes and strongly emphasizes that the Examiner should be aware that using a virtual machine for interpreting applets within an embedded data-processing system is fully known to any person of ordinary skill in the corresponding art since 1996, as quoted with reference to the Tim LINDHOLM and Frank YELLIN publication at page 2, lines 30 to 55 of the specification translation, while the documentation edited by SUN MICROSYSTEMS Inc. on the JAVACARD 2.1 Virtual Machine Specification was available to every body since March 1999, as quoted at the paragraph spanning pages 2 and 3 of the specification translation.

That the inventor had possession of the claimed invention which is not described in such a way to reasonably convey to one skilled in the art to embody the invention is traversed.

In the absence of evidences given by the Examiner, no evidences are given that the invention was actually not reduced to practice by the inventor(s), the objection is unfounded to the Applicant.

Claims 1 to 27 are rejected under 35 USC 112 second paragraph for they are generally narrative and indefinite.

Amended claims are now recasted to comply with the US practice.

Claims 1-7, 15-19 and 26-27 are rejected under 35 USC 102(b) for they lack novelty over U.S. patent 5,748,964 to Gosling.

Claims 7, 19, 21 and 23 have been cancelled without prejudice.

Although the patent to Gosling is said to meet the object of the invention, Gosling does not perform the verifying method of a fragment project as the method of the invention does.

Particularly, Applicant refers to the specification translation at page 5 from lines 7 to 29 in which the mode of operation of the system as disclosed by the U.S. patent 5,748,964 to Gosling is fully acknowledged and referred to as the third solution, known from the prior art.

Applicant also refers to the International Preliminary Examination Report as established by the International Preliminary Examination Authority and the official translation thereof, of which a copy is provided herewith.

The Examiner's attention should be drawn and made aware of that the object code verifier as disclosed by Gosling, referred to as D1, has the disadvantage of a complex and costly static code verification process both in terms of the code size required to control the processor and in terms of the RAM memory size, as well as in terms of calculation time, with these memory requirements being far greater than the resource capacity of most existing embedded (on-board) computer systems.

In contradistinction to the prior art solution, the invention makes use of a method for standardizing an original object code into a standardized object code with an empty stack branch instruction using typed registers unlike the prior art methods, in which the stack type at every branching target must be stored in memory. The verification method of the invention requires only the type of the execute stack during the instruction execution being verified and does not store the stack type in memory for other subprograms. As a result, the memory capacity requirement is significantly reduced.

More particularly with reference to the substitute specification translation at page 48 line 10 to page 49 line 2, the Applicant further emphasizes that the invention is directed to a novel technique for byte code verification of JAVACARD program fragments, designated as applets, or for program fragments for similar environments.

Basically the verification operation essentially consists in requiring that:

A) the virtual operand stack be empty at each target on a branching instruction, the program fragment being thus rejected if this constraint is not satisfied;

B) the type of the local variables, designated as registers, be identical at all point within a program fragment, designated as a method, the program fragment being thus rejected if this constraint is not satisfied.

Satisfying the above mentioned constraints, in accordance with the method of the invention, allows a very efficient embedded bytecode verifier to be implemented.

As clearly quoted on the preceding highlighted paragraph of the substitute specification translation, for a given program fragment using a maximum stack size of T_p and P_r registers, the memory size required by a byte code verifier according to the invention is a direct proportion to $T_p + P_r$.

By contrast, the verifiers known from the prior art, particularly from the US patent to Gosling, would have required a memory size in a direct proportion to $(T_p + P_r) \times N_b$, the product

of $T_p + P_r$ and the number N_b of targets of branching instructions included within the program fragment.

The invention also concerns a method for transforming any program fragment accepted by any prior art verifier into a program fragment accepted by the bytecode verifier of the invention.

Consequently, while most existing or marketed program fragments or applets would possibly be rejected when submitting them to a verifier of the invention, for these existing or marketed program fragments would not necessarily satisfy the above mentioned constraints A) and B), the invention also implements the method for transforming any existing or marketed program fragment, to be verified and then executed in accordance with the method for verifying of the invention.

The invention, as implemented, appears thus fully useful for any existing or marketed program fragment and does not offend against 35 USC 101 requirements, as contended by the Examiner.

The amended claims are recasted in accordance with the preceding statement by:

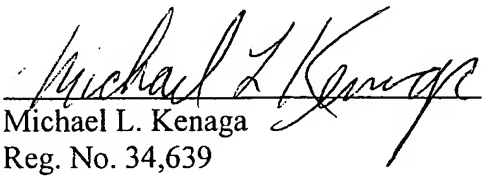
- canceling claims 1 to 3, 7, 19, 21 and 23;
- redrafting claim 4 by emphasizing corresponding constraint A) and B) which were explained before.

Remaining amended claims 4, 5, 6, 8-18, 20, 22, 24-27 are thus clearly not anticipated by Gosling and are patentable, since introducing the above discussed constraints so as to allow a significant reduction of the memory size over the most prominent bytecode verifiers of the prior art, particularly these developed by SUN MICROSYSTEMS as disclosed by Gosling, was not known or obvious at the date at which the invention was made;

- recasting claims 20, 22, 24 and 25 as independent claims to cancel reference to another claim of different category.

In view of the foregoing comments and amendment, reconsideration and allowance are requested.

Respectfully submitted,



Michael L. Kenaga
Reg. No. 34,639

DLA PIPER RUDNICK GRAY CARY US LLP
P.O. Box 64807
Chicago, Illinois 60664-0807
Phone: (312) 368-4000
Customer No.: 28465

ABSTRACT OF THE DISCLOSURE

A method and system for transforming and verifying downloaded programs fragments with data type restriction in an embedded system in which a program fragment being temporarily stored a verification process of the stored program fragment object code is executed instruction by instruction so as to discriminate for each instruction the existence of a target, a branching instruction target, a target of an exception handler call or a target of a subroutine call. On the occurrence of a target of a branching instruction as the current instruction, the empty status of the stack is verified and the program fragment is rejected otherwise. A verification process and updating of the effect of the current instruction on the data types of the type stack and the table of register types is performed. The verification process is successfully executed instruction by instruction until the table of register types is stable, with no modification being present, and interrupted with the program fragment being rejected otherwise.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier
Serial No.: 10/069,670
Filed: February 22, 2002
Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM
Group Art Unit: 2145
Examiner: SWEARINGEN, Jeffrey R.
Conf. No.: 3622
Customer No.: 28465
Our Ref.: P6451 (218728-000123)

Enclosures:
1. Transmittal Letter (2 pages)
2. Supplemental Amendment, including Abstract sheet attached thereto (31 pages)
3. Return receipt postcard

Mailed via First Class Mail on October 31, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: LEROY, Xavier
Serial No.: 10/069,670
Filed: February 22, 2002
Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM
Group Art Unit: 2145
Examiner: SWEARINGEN, Jeffrey R.
Conf. No.: 3622
Customer No.: 28465
Our Ref.: P6451 (218728-000123)

Enclosures:
1. Transmittal Letter (2 pages)
2. Supplemental Amendment, including Abstract sheet attached thereto (31 pages)
3. Return receipt postcard

Mailed via First Class Mail on October 31, 2005

TO:Toni Sousa COMPANY:

UNITED STATES PATENT AND TRADEMARK OFFICE

Facsimile Transmission

To:	Name:	Toni Sousa
	Company:	
	Fax Number:	916175236850
	Voice Phone:	
From:	Name:	Jeffrey R. Swearingen
	Voice Phone:	571-272-3921

37 C.F.R. 1.6 sets forth the types of correspondence that can be communicated to the Patent and Trademark Office via facsimile transmissions. Applicants are advised to use the certificate of facsimile transmission procedures when submitting a reply to a non-final or final Office action by facsimile (37 CFR 1.8(a)).

Fax Notes:

10/069,670

Date and time of transmission: Tuesday, August 22, 2006 11:40:46 AM
Number of pages including this cover sheet: 04

TO:Toni Sousa COMPANY:

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/069,670	02/22/2002	Xavier Leroy	P-6451	3622

28465 7590 01/23/2006

DLA PIPER RUDNICK GRAY CARY US LLP
P. O. BOX 64807
CHICAGO, IL 60664-0807

EXAMINER	
SWEARINGEN, JEFFREY R	

ART UNIT	PAPER NUMBER
2145	

DATE MAILED: 01/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

TO: Toni Sousa COMPANY:

Notice of Non-Compliant Amendment (37 CFR 1.121)	Application No. 10/069,670	Applicant(s) LEROY, XAVIER	
	Examiner Jeffrey R. Swearingen	Art Unit 2145	

– The MAILING DATE of this communication appears on the cover sheet with the correspondence address –

The amendment document filed on 03 November 2005 is considered non-compliant because it has failed to meet the requirements of 37 CFR 1.121. In order for the amendment document to be compliant, correction of the following item(s) is required.

THE FOLLOWING MARKED (X) ITEM(S) CAUSE THE AMENDMENT DOCUMENT TO BE NON-COMPLIANT:

- ☒ 1. Amendments to the specification:
  - ☐ A. Amended paragraph(s) do not include markings.
  - ☐ B. New paragraph(s) should not be underlined.
  - ☒ C. Other See Continuation Sheet.
- ☐ 2. Abstract:
  - ☐ A. Not presented on a separate sheet. 37 CFR 1.72.
  - ☐ B. Other _____.
- ☐ 3. Amendments to the drawings:
  - ☐ A. The drawings are not properly identified in the top margin as "Replacement Sheet," "New Sheet," or "Annotated Sheet" as required by 37 CFR 1.121(d).
  - ☐ B. The practice of submitting proposed drawing correction has been eliminated. Replacement drawings showing amended figures, without markings, in compliance with 37 CFR 1.84 are required.
  - ☐ C. Other _____.
- ☐ 4. Amendments to the claims:
  - ☐ A. A complete listing of all of the claims is not present.
  - ☐ B. The listing of claims does not include the text of all pending claims (including withdrawn claims)
  - ☐ C. Each claim has not been provided with the proper status identifier, and as such, the individual status of each claim cannot be identified. Note: the status of every claim must be indicated after its claim number by using one of the following status identifiers: (Original), (Currently amended), (Canceled), (Previously presented), (New), (Not entered), (Withdrawn) and (Withdrawn-currently amended).
  - ☐ D. The claims of this amendment paper have not been presented in ascending numerical order.
  - ☐ E. Other: _____.

For further explanation of the amendment format required by 37 CFR 1.121, see MPEP § 714 and the USPTO website at http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/officeflyer.pdf .

TIME PERIODS FOR FILING A REPLY TO THIS NOTICE:

1. Applicant is given **no new time period** if the non-compliant amendment is an after-final amendment or an amendment filed after allowance. If applicant wishes to resubmit the non-compliant after-final amendment with corrections, the **entire corrected amendment** must be resubmitted within the time period set forth in the final Office action.

2. Applicant is given **one month**, or thirty (30) days, whichever is longer, from the mail date of this notice to supply the **corrected section** of the non-compliant amendment in compliance with 37 CFR 1.121, if the non-compliant amendment is one of the following: a preliminary amendment, a non-final amendment (including a submission for a request for continued examination (RCE) under 37 CFR 1.114), a supplemental amendment filed within a suspension period under 37 CFR 1.103(a) or (c), and an amendment filed in response to a *Quayle* action.

   Extensions of time are available under 37 CFR 1.136(a) only if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action.

   Failure to timely respond to this notice will result in:
   Abandonment of the application if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action; or
   Non-entry of the amendment if the non-compliant amendment is a preliminary amendment or supplemental amendment.

TO:Toni Sousa COMPANY:

Continuation Sheet (PTOL-324) **Application No. 10/069,670**

Continuation of 1(c) Other: The substitute specification was not included in the submissions of 11/3/2005 or 10/17/2005.

JASON CARDONE
SUPERVISORY PATENT EXAMINER

Assignee: Trusted Logic
Inventor: Xavier Leroy
Serial No.: 10/069,670
Title: A METHOD FOR TRANSFORMING AND VERIFYING DOWNLOADED PROGRAM FRAGMENTS WITH DATA TYPES RESTRICTIONS AND CORRESPONDING SYSTEM

Holland & Knight LLP
10 St. James Avenue
11TH Floor
Boston, MA 02116

DECLARATION OF OLGA MELENDEZ

1. I, Olga Melendez, am the docketing clerk of the Chicago office of DLA Piper Rudnick Gray Cary US LLP, being located at 203 North LaSalle Street, Suite 1900 Chicago, Illinois 60601-1293.

2. I was informed that U.S. Patent Application Number 10/069,670 went abandoned due to a failure to respond to a Notice of Non-Compliant Amendment (37 CFR 1.121). The Notice of Non-Compliant Amendment was mailed from the U.S. Patent and Trademark Office to our office on January 23, 2006.

3. All correspondence that is received by this office is tracked within our docketing system.

4. Upon learning that U.S. Patent Application Number 10/069,670 went abandoned, I searched our docketing system and found no indication that such Notice of Non-Compliant Amendment was ever received by our office.

5. Further, I have not found such Notice of Non-Compliant Amendment after a physical search of our office.

6. Accordingly, it is my belief that such Notice of Non-Compliant Amendment was never received by DLA Piper Rudnick Gray Cary US LLP.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Olga L. Melendez
Olga Melendez

10-23-06
Date

4082362_v1

100 ans

CABINET PLASSERAUD
Depuis 1906
European Patent & Trademark Attorneys
Conseils en propriété industrielle

Paris
65 / 67 rue de la Victoire
75440 PARIS CEDEX 09
Tél. +33 (0) 1 40 15 70 00
Fax +33 (0) 1 42 00 01 59
info@plasse.com
www.plasse.com

ASSOCIÉS
J.J.A. M.-H. JACQUELIN
J.J.A. G. KOCH
J.J.A. D. BOULINGUIEZ
J.J.A. F. BERGDIN
J.J.A. B. LOISEL
J.A. E. BURBAUD
J.A. G. KIESEL LE COGUIEC
J.A. C. NARGOLWALLA
J.J.A. S. VERDURE

COLLABORATEURS
J.J.A. J.-M. GORREE
J.A. B. THOMAS
J.J.A. C. TOUATI
J.A. B. DEJARDINS
J.A. H. GAREL
J.A. G. RINGEISEN
J.A. D.-I. TAYER
J. P. BOIRE
J.A. A. HASSINE
J. D. ATTALI
J.A. C. VOLIGNY
J.A. P. BARBIER
G. COUSIN
G. DUROUT
A. BASTIEN
J.A. P. BOYLE
S. MILLET
J. F. THIEL
G. SIMON
A.-F. LE HOLLE
S. LIMOGES
J. C. DE MARGUIL VILLETTE
G. BOSSARD
J. P. PRIFUR
I. SCHREIBER
J.A. G. PERIN
T. COUAILLET
J.J.A. M. FRECHEDE
J.A. E. SEINAYE
G. DE RIVOYRE
J. R. CEFAI
D. RAETS
F. NIEMANN
N. RICHARD
J.A. D. PÖPPING
C. VERSCHELDE
E. MASSE
J.J.A. E. RENARD
P. TOURAT
J. S. VIGNESOULT
S. ANDRAUD
P. DUPIRE

SECRÉTAIRE GÉNÉRAL
A. LEREDUF

ADMINISTRATION
C. HEME de LACOTTE
E. DEMERY (Brevets)
J.A. E. BENSUSSAN (Marques)
M. PICCOLI (Marques)
J. C. PAVOT (Annuités)

Lyon
27 rue de la Villette
69003 LYON
Tél. +33 (0) 4 37 91 62 70
Fax +33 (0) 4 37 91 62 79
contact-lyon@plasse.com
www.plasse.com

COLLABORATEURS
J.J.A. R. FLEURANCE
J. F. COGNIAT
J.A. S. PICARD
J.A. O. VERMANDER
E. BEAUVAIR
J. N. JOUANNIC

J. Conseil en propriété industrielle
J. Patent Attorney
J. Trade mark Attorney
J. US Patent Attorney

Via facsimile
0 00 1 603 668 2970
+ MAIL CONFIRMATION

GROSSMAN, TUCKER,
PERREAULT & PFLEGER, PLLC
55 South Commercial Street
MANCHESTER
NH 03101
USA

RECEIVED WITH THANKS

To the attention of Mr. Brian J. COLANDREO

O/Ref.: MF/AD – US 10/069,670

Paris, February 9, 2006

US Patent Application No. 10/069,670
National Phase of PCT/FR00/02349 filed August 21, 2000
In the name of X. Leroy, assignment to TRUSTED LOGIC
Power of attorney duly signed

Dear Brian·

I refer to the last e-mail sent to you by Bertrand LOISEL and feel very pleased to send you back the Power of Attorney duly signed by Dominique BOLIGNANO the Chief Executive Officer of my client Trusted Logic.

Please, take the necessary steps to have this Power of Attorney duly enforced with the US Patent and Trademark Office and the file transferred to your office from:

DLA PIPER RUDNICK GRAY CARY US LLP
203 North LaSalle Street, Suite 1900
Chicago, ILLINOIS 60601-1293

Looking forward to receiving soon comments in this respect where appropriate,

I remain.

Very truly yours.

Michel FRECHEDE

PLEASE ACKNOWLEDGE RECEIPT OF THESE INSTRUCTIONS BY RETURN

Encl.: Power of Attorney

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.:	10/069,670	§
Inventor:	LEROY, Xavier	§
Filing Date:	02/22/2002	§

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

REVOCATION OF POWER OF ATTORNEY WITH NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

I hereby revoke any previous Powers of Attorney given in the above-identified application, and appoint the practitioners associated with Customer Number **32047** with full power of substitution and revocation, to prosecute this application, to make alterations and amendments therein, to transact all business in the Patent and Trademark Office connected therewith, to receive any Letters Patent, and to file any request for a certificate of correction that may be deemed appropriate.

Please change the correspondence address to the address associated with Customer Number **32047**.

STATEMENT UNDER 37 C.F.R. 3.73(b)

Pursuant to 37 C.F.R. § 3.73, TRUSTED LOGIC, 5 rue du Bailliage, 78000 Versailles, France, states that it is the Assignee of the entire right, title, and interest in the above-identified patent application by virtue of an assignment:

☐ Filed concurrently herewith for recording, a copy of which is attached hereto.

☒ Previously recorded on
Reel 012858
Frame 0972

The undersigned (whose title is supplied below) is the representative for the Assignee of the entire right, title, and interest in the patent application identified above, and is authorized to act on behalf of the Assignee.

TRUSTED LOGIC

By: Dominique BOLIGNANO

Title: PDG

PLS021
PLS017

DLA PIPER RUDNICK GRAY CARY

DLA Piper Rudnick Gray Cary US LLP
203 North LaSalle Street, Suite 1900
Chicago, Illinois 60601-1293
T 312.368.4000
F 312.236.7516
W www.dlapiper.com

R. BLAKE JOHNSTON
Blake.johnston@dlapiper.com
T 312.368.4000 F 312.236.7516

February 16, 2006

VIA UPS 2ND DAY DELIVERY

DOCKETED
FEB 16 2006
PIPER RUDNICK

Brian J. Colandreo, Esq.
GROSSMAN, TUCKER,
PERREAULT & PFLEGER, PLLC
55 S. Commercial Street
Manchester, NH 03101

Re: Cabinet Plasseraud – Request to Transfer File
US National Phase Appl. No. 10/069,070
Title: Management Protocol, Method for Verifying and Transforming a Downloaded Program Fragment and Corresponding Systems
In the name of Trusted Logic
Our Reference No. 218728-000123 (P6451)

Dear Mr. Colandreo:

In accordance with the February 9, 2006, request of Mr. Michel Frechede of Cabinet Plasseraud, we have been advised that you are assuming responsibility for the above referenced matter. As instructed, we are transferring the original prosecution file DLA Piper Rudnick Gray Cary US LLP currently maintains for this pending patent application.

As a courtesy, we are enclosing a printout from our docketing database indicating all the relevant data for this application. However, we recommend you rely solely on the information in the file when entering into your docket system.

Since we are no longer responsible for this matter, we will be removing it from our docket maintenance system. If we receive any correspondence relating to this application in the near future, we will promptly forward same to your attention.

DLA PIPER RUDNICK GRAY CARY

Kindly acknowledge receipt of the enclosed file by signing and returning the enclosed duplicate of this correspondence.

Very truly yours,

DLA PIPER RUDNICK GRAY CARY US LLP

R. Blake Johnston

RBJ/ib
Enclosures

cc: Bertrand Loisel
Michel Frechede

Receipt of the foregoing is hereby acknowledged.

By: Dana Robertson Date: 3/30/06

-CHGO1:30755188.v1

Holland & Knight

Tel 617 523 2700
Fax 617 523 6850

Holland & Knight LLP
10 St. James Avenue
Boston, MA 02116
www.hklaw.com

TO:		
Jeffrey R. Swearingen **Group Art Unit 2145**	U.S. Patent and Trademark Office	(571) 273-8300 (571) 273-3921
NAME	COMPANY/FIRM	FAX NUMBER
Alexandria	VA	
CITY	STATE	(TELEPHONE NUMBER)

FROM:		
Brian J. Colandreo, Reg. No. 42,427	(617) 305-2143	3
NAME	TELEPHONE	TOTAL PAGES (Including Cover Sheet)

FOR THE RECORD:		
DATE: August 17, 2006	URGENCY: ☐ SUPER RUSH	☒ RUSH ☐ REGULAR
FAXED BY: Toni Sousa	FILE #:	CLIENT NAME:
CONFIRMED: ☐ YES ☐ NO	NAME:	TIME:

If you did not receive all of the pages or find that they are illegible, please call **617 523 2700**

MESSAGE:

Dear Examiner Swearingen:

As requested, attached is a Revocation of Power of Attorney With New Power of Attorney form regarding U.S. Application No. 10/069,670, filed 22 February 2002.

As soon as you receive this form please fax over a copy of the action dated 23 January 2006 to (617) 523-6850.

Thank you for your assistance in this matter.

Respectfully submitted,

Brian J. Colandreo
Reg. No. 42,427

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.:	10/069,670	§
Inventor:	LEROY, Xavier	§
Filing Date:	02/22/2002	§

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

REVOCATION OF POWER OF ATTORNEY WITH NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

I hereby revoke any previous Powers of Attorney given in the above-identified application, and appoint the practitioners associated with Customer Number **32047** with full power of substitution and revocation, to prosecute this application, to make alterations and amendments therein, to transact all business in the Patent and Trademark Office connected therewith, to receive any Letters Patent, and to file any request for a certificate of correction that may be deemed appropriate.

Please change the correspondence address to the address associated with Customer Number **32047**.

STATEMENT UNDER 37 C.F.R. 3.73(b)

Pursuant to 37 C.F.R. § 3.73, TRUSTED LOGIC, 5 rue du Bailliage, 78000 Versailles, France, states that it is the Assignee of the entire right, title, and interest in the above-identified patent application by virtue of an assignment:

☐ Filed concurrently herewith for recording, a copy of which is attached hereto.

☒ Previously recorded on
Reel 012858
Frame 0972

The undersigned (whose title is supplied below) is the representative for the Assignee of the entire right, title, and interest in the patent application identified above, and is authorized to act on behalf of the Assignee.

TRUSTED LOGIC

By: Dominique Bolignano

Title: PDG

Actions Due

Wednesday, December 20, 2006 Page: 1

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 23-Apr-2003

Action Type: Status Response sent date: 23-May-2005

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
1st Office Action Received?	01-Jul-2005	Reminder	23-May-2005
Status Follow Up Date	23-Nov-2005	Due Date	23-May-2005

Remarks:

4/29/05: Per Examiner Jeffrey R. Swearingen (AU 2145 571-272-3921) - 2 more months.
5/23/05: Received from USPTO - Office Action Summary

Created By: sc12551

User ID: warnes

Date Created: 22-Feb-2002

Last Update: 24-May-2005

Actions Due

Wednesday, December 20, 2006 Page: 1

Log Number: 6451 **SubCase:** **Resp. Office:** CH

Country: US United States of America **Case Type:** ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS **Client.Matter:** 218728-000123

Nick Name: **Filing Date:** 22-Feb-2002

Status: Transferred **Application # :** 10/069670 **Base Date:** 22-Feb-2002

Action Type: Application Status Check **Response sent date:**

Primary Atty: **Resp. Atty 1:** **Resp. Atty 2:** (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
Application Status Check	22-Aug-2003	Due Date	22-Feb-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 14-May-2002

Last Update: 14-May-2002

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 22-Feb-2002

Action Type: Foreign Filing Response sent date:

Primary Atty: Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
Foreign Filing Reminder	22-Aug-2002	Due Date	22-Feb-2002
Foreign Filing Due	22-Feb-2003	Final	22-Feb-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 14-May-2002

Last Update: 14-May-2002

Actions Due

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS

Client Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 22-Jul-2002

Action Type: IDS Response sent date:

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
IDS (originally due 5/22/02)	22-Jul-2002	Due Date	23-Sep-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 28-Jun-2002

Last Update: 23-Sep-2002

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS

Client.Matter: 218728-000123

Nick Name:

Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 23-Apr-2003

Action Type: Status Response sent date: 23-May-2005

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
1st Office Action Received?	01-Jul-2005	Reminder	23-May-2005
Status Follow Up Date	23-Nov-2005	Due Date	23-May-2005

Remarks:

4/29/05: Per Examiner Jeffrey R. Swearingen (AU 2145 571-272-3921) - 2 more months.
5/23/05: Received from USPTO - Office Action Summary

Created By: sc12551

User ID: warnes

Date Created: 22-Feb-2002

Last Update: 24-May-2005

Actions Due

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 29-Sep-2005

Action Type: Status Response sent date:

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
Follow-up to 9/29/05 Response	29-Mar-2006	Reminder	

Remarks:

Created By: IB11623

User ID: IB11623

Date Created: 10-Jan-2006

Last Update: 10-Jan-2006

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 23-Jun-2002

Action Type: STATUS HAVE WE REC'D FILING RE Response sent date:

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
STATUS HAVE WE REC'D FILING	23-Jun-2002	Due Date	10-May-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 22-Feb-2002

Last Update: 14-May-2002

Actions Due

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 23-Aug-1999

Action Type: Substantive Examination Response sent date:

Primary Atty: Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
Request for Examination	23-Aug-2002	Due Date	22-Feb-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 14-May-2002

Last Update: 14-May-2002

Actions Due

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 22-Feb-2002

Action Type: Substantive Examination Response sent date:

Primary Atty: Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
Request for Examination	22-Feb-2005	Due Date	22-Feb-2002

Remarks:

Created By: ib11623

User ID: ib11623

Date Created: 14-May-2002

Last Update: 14-May-2002

Actions Due

Log Number: 6451 SubCase: Resp. Office: CH

Country: US United States of America Case Type: ORD

Title: MANAGEMENT PROTOCOL, METHOD FOR VERIFYING AND TRANSFORMING A DOWNLOADED PROGRAMME FRAGMENT AND CORRESPONDING SYSTEMS Client.Matter: 218728-000123

Nick Name: Filing Date: 22-Feb-2002

Status: Transferred Application # : 10/069670 Base Date: 16-May-2005

Action Type: US-Non-Final Office Action Response sent date: 29-Sep-2005

Primary Atty: MLK Resp. Atty 1: Resp. Atty 2: (•) Attorney ( ) Paralegal

Action(s) Due	Due Date	Indicator	Taken
2 mon: Resp to OA due	16-Jun-2005	Reminder	29-Sep-2005
1 mon: Resp to OA due	16-Jul-2005	Reminder	29-Sep-2005
Resp to Office Act due today	16-Aug-2005	Due Date	29-Sep-2005
Rsp to Office Action - 1st Ext	16-Sep-2005	Due Date	29-Sep-2005
Rsp to Office Action - 2nd Ext	16-Oct-2005	Due Date	29-Sep-2005
Return Postcard Rec'd	29-Oct-2005	Due Date	31-Oct-2005
Rsp to Office Action - Fnl Ext	16-Nov-2005	Final	29-Sep-2005

Remarks:

8/3/05: Letter from Client via courier - enclosing comments and instructions for response to OA due 8/16/05

9/20/05: Letter from Client: inquiring if response was filed before 8/16/05

Created By: warnes

User ID: rs18195

Date Created: 24-May-2005

Last Update: 31-Oct-2005